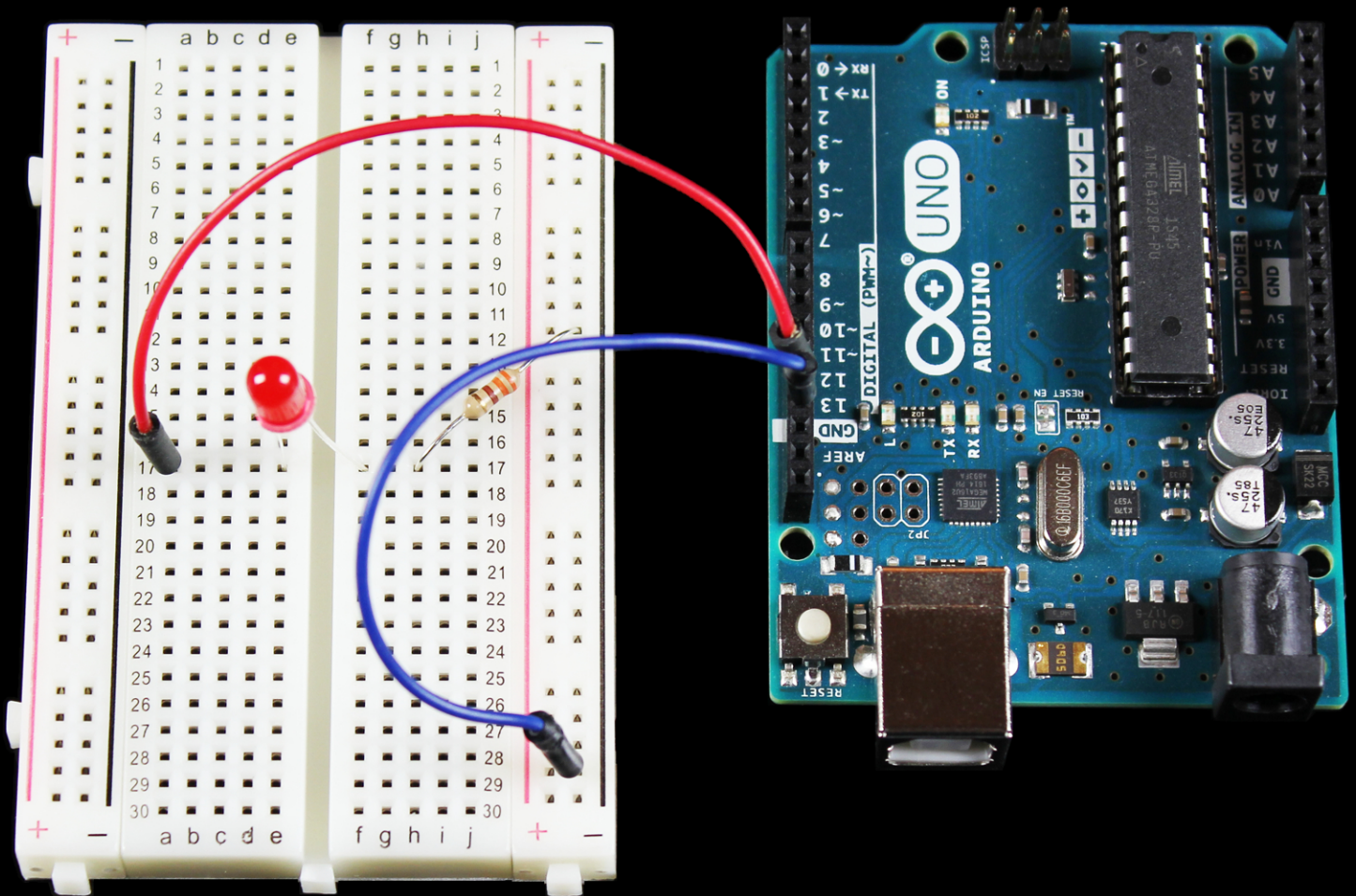


Arduino For Beginners

Step-By-Step Guide To Getting Started



Makerspaces
.com

Andrew Miller

Arduino For Beginners
Step-By-Step Guide To Getting Started

Copyright © 2017 Makerspaces.com

ISBN-10: 0-9979584-2-1

ISBN-13: 978-0-9979584-2-3

Published by Makerspaces.com

P.O Box 4147

Clearwater, FL 33758

www.makerspaces.com

Author: Andrew Miller

Edition: 1

Disclaimer - The information and projects in this book are distributed on an “as is” basis, without warranty. While every precaution has been taken when preparing the projects and information, the author and publisher shall not have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this book. The author and publisher also assume no responsibility for errors or omissions in this book. There are small parts and electricity involved in some of the projects and because of this, supervision is recommended for small children who attempt to create these projects.

This work is published under a Creative Commons license:

Attribution—NonCommercial-ShareAlike 4.0 International License

(CC BY-NC-SA 4.0)

<https://creativecommons.org/licenses/by-nc-sa/4.0/>



About The Author



Andrew Miller is the Founder and CEO of Makerspaces.com which he started in 2014 to help schools and libraries learn more about creating a successful makerspace. He is a strong believer in maker education and hands-on learning as a way to help students acquire the skills needed to succeed in the 21st century. He comes from a long line of teachers and is committed to helping improve the educational system through maker education. Andrew has been a maker since he was 8 years old and hopes to inspire others to find the joy in making.

To my daughter Lilly –

I can't wait for the day when you're old enough to create things in our
makerspace. This book was written for you.



Table of Contents

1. Introduction.....	7
a. Projects In This Book	8
b. Materials Needed	9
c. Tools Needed	10
d. Workspace & Storage	17
2. Intro To Basic Electronics	19
a. Electronic Components	20
b. What is a Circuit?	26
c. Schematic Diagrams	27
d. How To Determine A Resistor Size	29
e. How To Use a Breadboard	30
f. Practice Project #1	32
g. Practice Project #2	35
3. Arduino Hardware Basics	39
a. What Is Arduino?	40
b. Types of Arduino Boards	41
c. What Can You Do With Arduino?	43
d. Arduino Uno Breakdown	45
e. Powering The Board	47
f. Arduino Shields	48
g. Arduino Sensors	49
4. Arduino Software Basics	50
a. Arduino Programming Software – IDE	52
b. Arduino Software Breakdown.....	53

5. Your First Arduino Project	55
6. Arduino Practice Projects	68
a. Download The Arduino Project Code (APC)	70
b. Connect & Configure The Arduino Uno	70
c. Project 1 – Test The Arduino	72
d. Project 2 – Blink An LED	74
e. Project 3 – Pushbutton	76
f. Project 4 – Potentiometer	78
g. Project 5 – Fade An LED	80
h. Project 6 – Scrolling LED	82
i. Project 7 – Bargraph	84
j. Project 8 – Multiple LEDs	86
k. Project 9 – RGB LED	88
l. Project 10 – Photoresistor	90
m. Project 11 – Temp Sensor	92
n. Project 12 – Tone Melody	94
o. Project 13 – Servo	96
p. Project 14 – Motor	98
q. Project 15 – LCD Screen	100
7. The Arduino Bit From littleBits®	102
8. Arduino Troubleshooting	112
9. How To Solder Your Arduino Projects	114
a. What Is Soldering?	115
b. Soldering Tools & Materials	116
c. Safety	122
d. How To Solder	123
e. How To Desolder	129
10. Arduino & Electronics Suppliers	131
11. Never Stop Learning	133

1

Introduction

Introduction

I wrote this book for the absolute beginner who has never even touched an Arduino or worked with electronics before. It was written in a very simple and easy to understand way. This book contains step-by-step instructions that will help you learn about the Arduino hardware, the software and will combine the two in order to create your first project. In fact, there are over 17 projects in this book that are easy to do and each will teach a specific concept.

Learning Arduino isn't hard and the skills you acquire are extremely valuable in the 21st century. Whether you're a teacher or librarian in a makerspace or a K-12 student, you can do this!

Projects In This Book

The projects in this book were designed to introduce you to Arduino while helping you learn about basic electronics. These projects utilize a breadboard in order for you to create the circuits without the need for soldering.

The following are the projects you will be able to create and each comes with a material list in addition to a step by step guide.

1. Test The Arduino
2. Blink An LED
3. Pushbutton
4. Potentiometer
5. Fade An LED
6. Scrolling LED
7. Bargraph
8. Multiple LEDs
9. RGB LEDs
10. Photoresistor
11. Temp Sensor
12. Tone Melody
13. Servo
14. Motor
15. LCD Screen

Materials Needed

In order to do the projects in this book, you will need some basic materials. The following items are easy to find online and are not expensive to buy.

- Arduino Uno Board
- USB A-to-B Cable
- Breadboard – half size
- Jumper Wires
- LED (5mm)
- RGB LED
- Temp Sensor
- Pushbutton Switch
- Potentiometer
- Photoresistor
- Servo
- DC Motor
- Piezo Buzzer
- LCD Screen
- NPN Transistor
- 10k Ohm Resistor
- 220 Ohm Resistor
- 330 Ohm Resistor

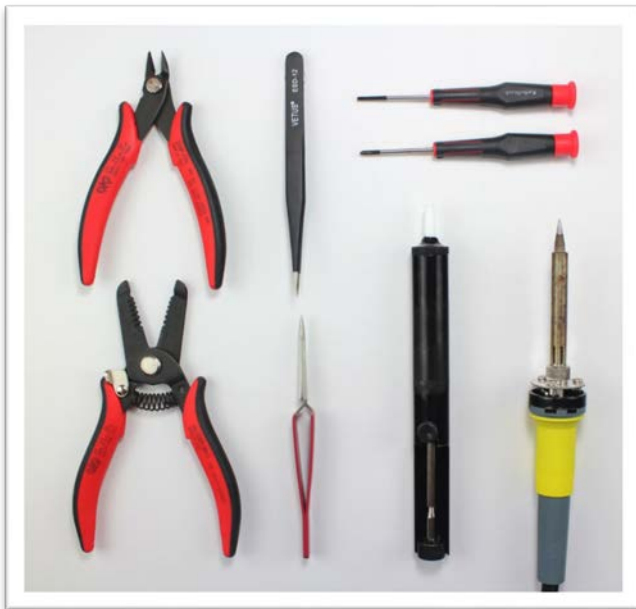


Tools Needed

The projects in this book were designed so that you wouldn't need a ton of tools to complete them. In fact, you really only need a few very basic hand tools which are listed below. The items under optional are tools you should buy if you are going to work on more complex Arduino and electronics projects.

Required

1. Needle-nose pliers
2. Wire Strippers/Cutters
3. Fine Tip Straight Tweeters
4. Test Leads (Alligator Clips)



Optional

1. Digital Multimeter
2. Soldering Iron
3. Solder Sucker
4. Panavise Jr
5. Flush Cutters
6. Battery Holders
7. Precision Screwdriver Set
8. Helping 3rd Hand
9. Heat Gun

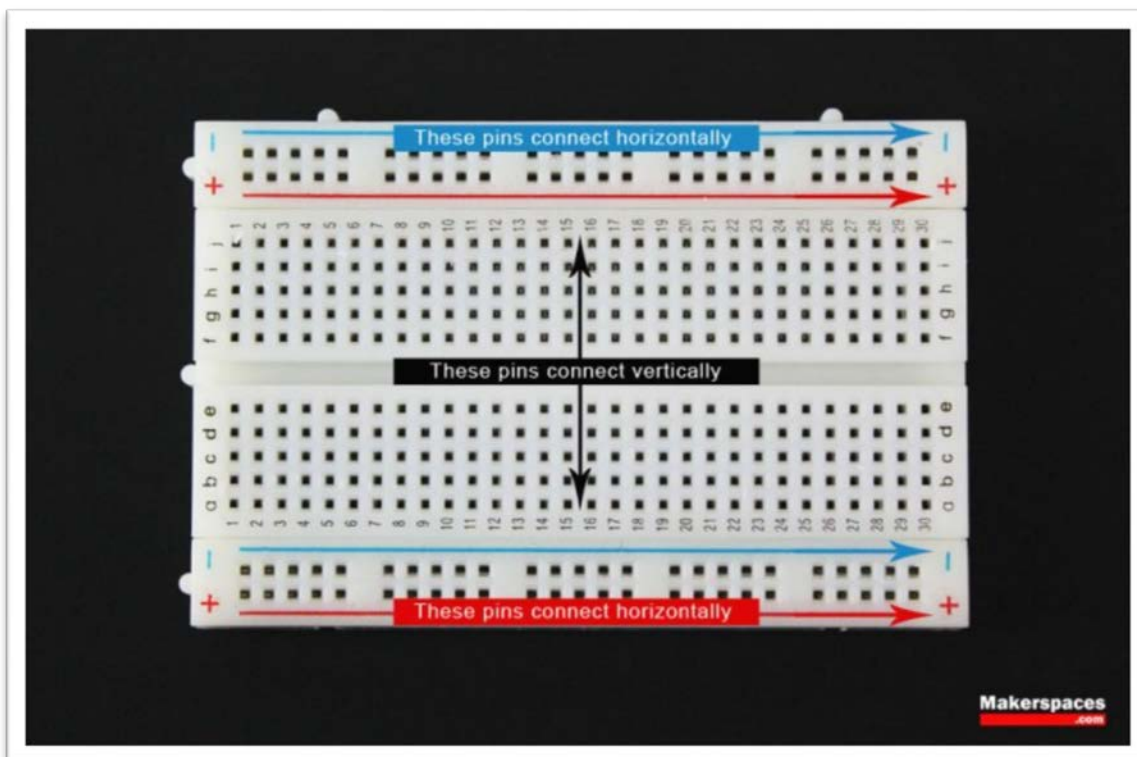
Here is a brief description of the required and optional tools that you'll be using. This isn't a complete list but it does outline the items you're most likely to need when working with Arduino and basic electronics.

Breadboard

One of the most important items you need when working with Arduino is a solderless breadboard. This device allows you to prototype your project without having to permanently solder the circuit together. Using a breadboard allows you to create temporary designs and experiment with different circuit configurations.

Inside the holes (tie points) of the plastic housing, are metal clips which are connected to each other by strips of conductive material. The holes in a breadboard are connected in rows both horizontally and vertically as shown below.

The breadboard is not powered on its own and needs power brought to it from the Arduino board, a battery or power pack.



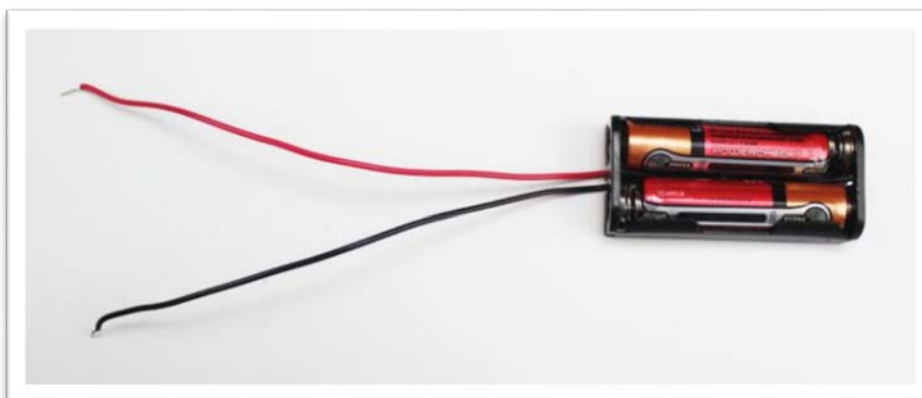
Digital Multimeter

A multimeter is a device that's used to measure electric current (amps), voltage (volts) and resistance (ohms). It's great for troubleshooting circuits and is capable of measuring both AC and DC voltage.



Battery Holders

A battery holder is a plastic case that holds batteries from 9V to AA. Some holders are enclosed and may have an on/off switch built in. These can be used to power a breadboard.



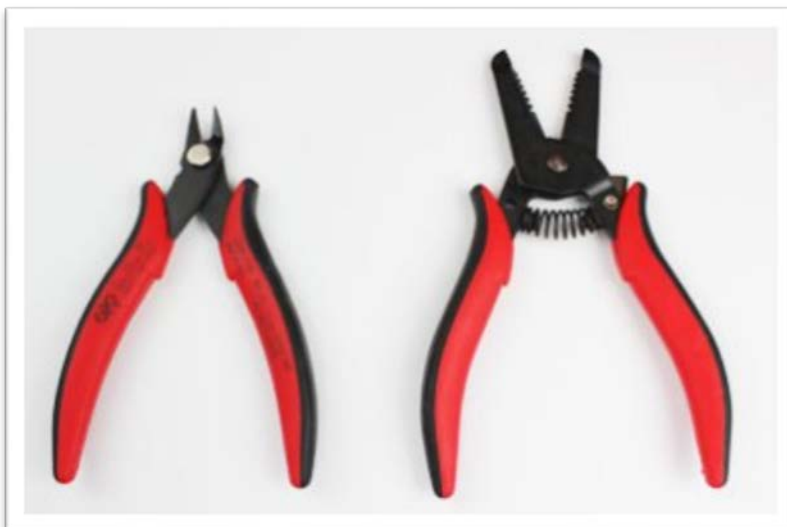
Test Leads (Alligator Clips)

Test leads are great for connecting components together to test a circuit without the need for soldering.



Wire Cutter

Wire cutters are essential for stripping stranded and solid copper wire of varying gauges.



Precision Screwdriver Set

Precision screwdrivers are also known as jeweler's screwdrivers and usually come as a set. The advantage of these over normal screwdrivers is the precision tips of each driver. These are very handy when working with electronics that contain tiny screws.



Helping 3rd Hand

When working with electronics, it seems you never have enough hands to hold everything. This is where the helping 3rd hand comes in. Great for holding circuit boards or wire when soldering.



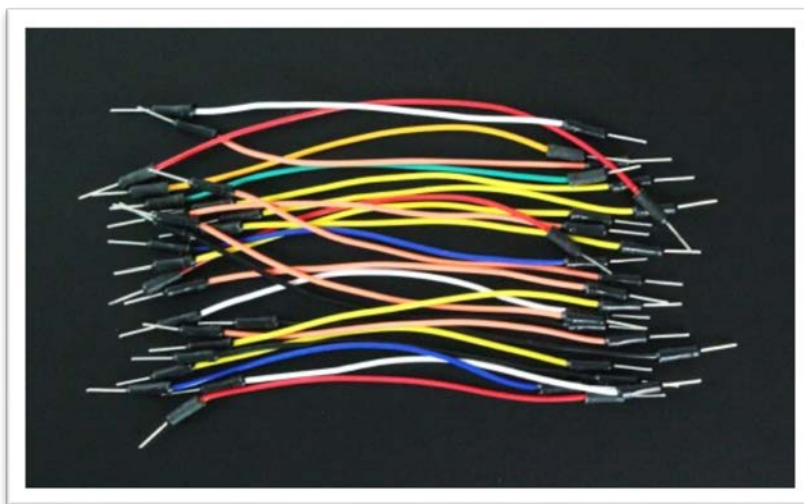
Heat Gun

A heat gun is used to shrink plastic tubing known as heat shrink to help protect exposed wire. Heat shrink has been called the duct tape of electronics and comes in handy in a wide variety of applications. Heat guns usually have high and low heat settings.



Jumper Wire

These wires are used with breadboard and development boards and are generally 22-28 AWG solid core wire. Jumper wires can have male or female ends depending on how they need to be used. They are used to connect components together and come in a variety of colors & lengths.



Soldering Iron

When it's time to create a permanent circuit, you'll want to solder the parts together. To do this, a soldering iron is the tool you would use. These irons can have fixed or variable heat control.



Soldering Station

A soldering station is a more advanced version of the stand-alone soldering iron. They usually have more precise temperature control and some come with a built-in iron holder.

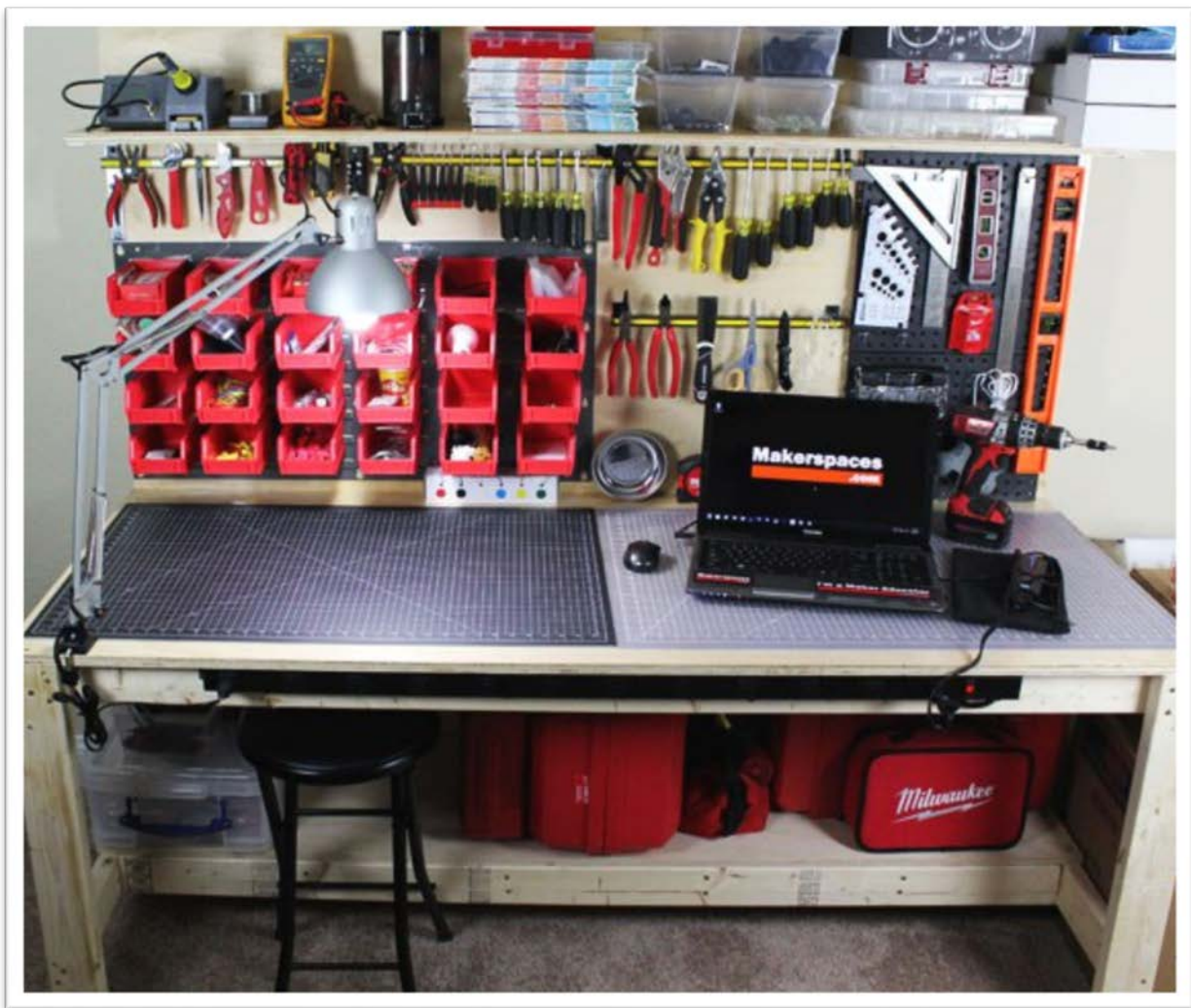


Workspace & Storage

Before you start working with Arduino and electronics you'll want to have a good workspace. There are a lot of small parts involved in these projects and having an organized workbench will save you headaches.

Build a Workbench

Your workbench doesn't need to be fancy and it doesn't even need to be store bought. We wrote a post on Makerspaces.com about building the below [workbench for under \\$100](#). Check it out if you want the step by step build plans.



Storage

Some electronic components can be as small as a pencil eraser so it's a good idea to keep everything organized. A popular storage option is to use clear plastic storage boxes for small parts. These storage boxes come in a variety of sizes and usually have dividers.

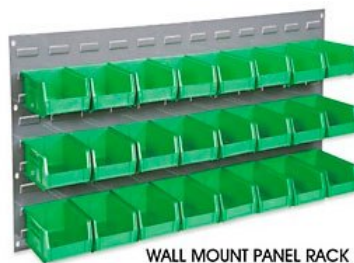


Another organization option is to use storage bins that hang from a wall mounted bar. The red bin as shown above can be mounted to a workbench or put in a cabinet. These are very handy for keeping all your small items close by but out of the way.

You can find these bins and racks at Uline.com



BENCH RACK



WALL MOUNT PANEL RACK



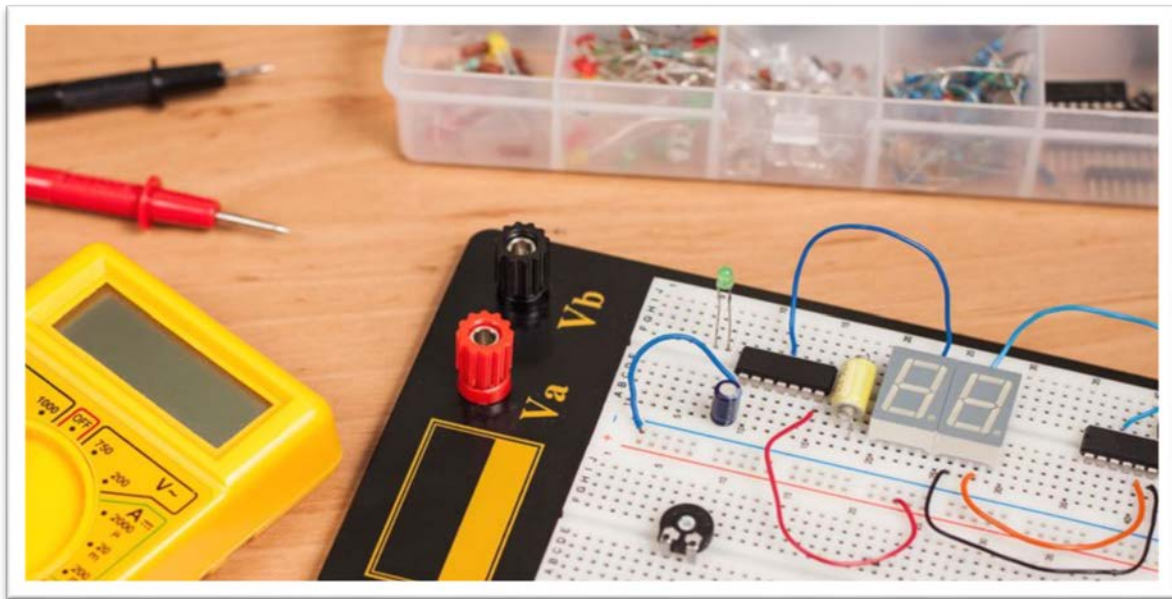
WALL MOUNT SINGLE RAIL

2

Intro To Basic Electronics

Intro To Basic Electronics

Before we start working with Arduino, you need to learn about the electrical components and principles that are commonly used in conjunction with these projects. If you already have an understanding of these components, feel free to jump ahead to the other sections on circuits or schematics.



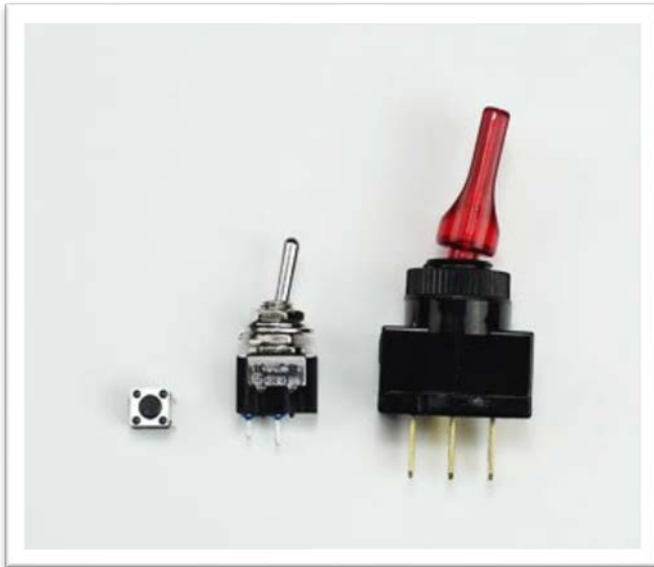
Electronic Components

There are literally hundreds of different electronic components available on the market. On the following pages you will get a quick breakdown of the most commonly used components that make your Arduino projects come to life.

If you're new to electronics, it can be very easy to get overwhelmed when seeing all these parts. It's a good idea to just get familiar with the names and their basic functions. Once you start building your practice projects it will be easier to understand these components.

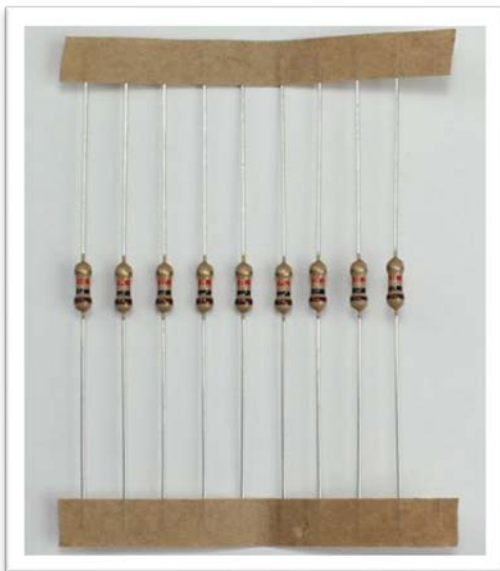
Switch

Switches can come in many forms such as pushbutton, rocker, momentary and others. Their basic function is to interrupt electric current by turning a circuit on or off.



Resistor

Resistors are used to resist the flow of current or to control the voltage in a circuit. The amount of resistance that a resistor offers is measured in Ohms. Most resistors have colored stripes on the outside and this code will tell you its value of resistance. You can use a multi-meter or Digikey's [resistor color code calculator](#) to determine the value of a resistor.



Variable Resistor (Potentiometer)

A variable resistor is also known as a potentiometer. These components can be found in devices such as a light dimmer or volume control for a radio. When you turn the shaft of a potentiometer the resistance changes in the circuit.



Light-Dependent Resistor (LDR)

A light-dependent resistor is also a variable resistor but is controlled by the light versus turning a knob. The resistance in the circuit changes with the intensity of the light. These are often found in exterior lights that automatically turn on at dusk and off at dawn.



Capacitor

Capacitors store electricity and then discharge it back into the circuit when there is a drop in voltage. A capacitor is like a rechargeable battery and can be charged and then discharged. The value is measured in F (Farad), nano Farad (nF) or pico Farad (pF) range.



Diode

A diode allows electricity to flow in one direction and blocks it from flowing the opposite way. The diode's primary role is to route electricity from taking an unwanted path within the circuit.



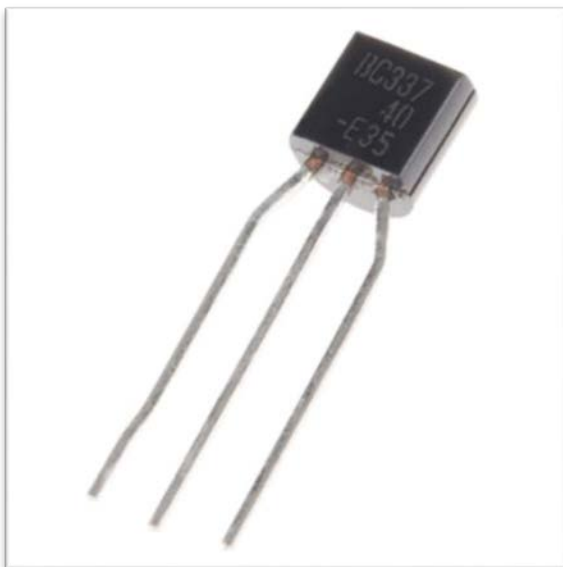
Light-Emitting Diode (LED)

A light-emitting diode is like a standard diode in the fact that electrical current only flows in one direction. The main difference is an LED will emit light when electricity flows through it. Inside an LED there is an anode and cathode. Current always flows from the anode (+) to the cathode (-) and never in the opposite direction. The longer leg of the LED is the positive (anode) side.



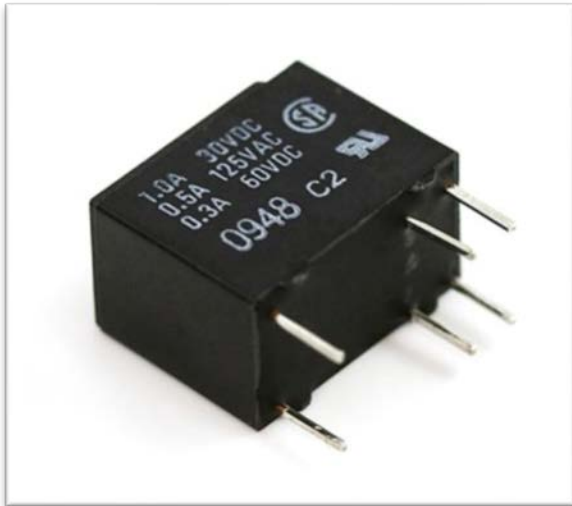
Transistor

Transistors are tiny switches that turn a current on or off when triggered by an electric signal. In addition to being a switch, it can also be used to amplify electronic signals. A transistor is similar to a relay except with no moving parts.



Relay

A relay is an electrically operated switch that opens or closes when power is applied. Inside a relay is an electromagnet which controls a mechanical switch.



Integrated Circuit (IC)

An integrated circuit is a circuit that's been reduced in size to fit inside a tiny chip. This circuit contains electronic components like resistors and capacitors but on a much smaller scale. Integrated circuits come in different variations such as 555 timers, voltage regulators, microcontrollers and many more. Each pin on an IC is unique in terms of its function.

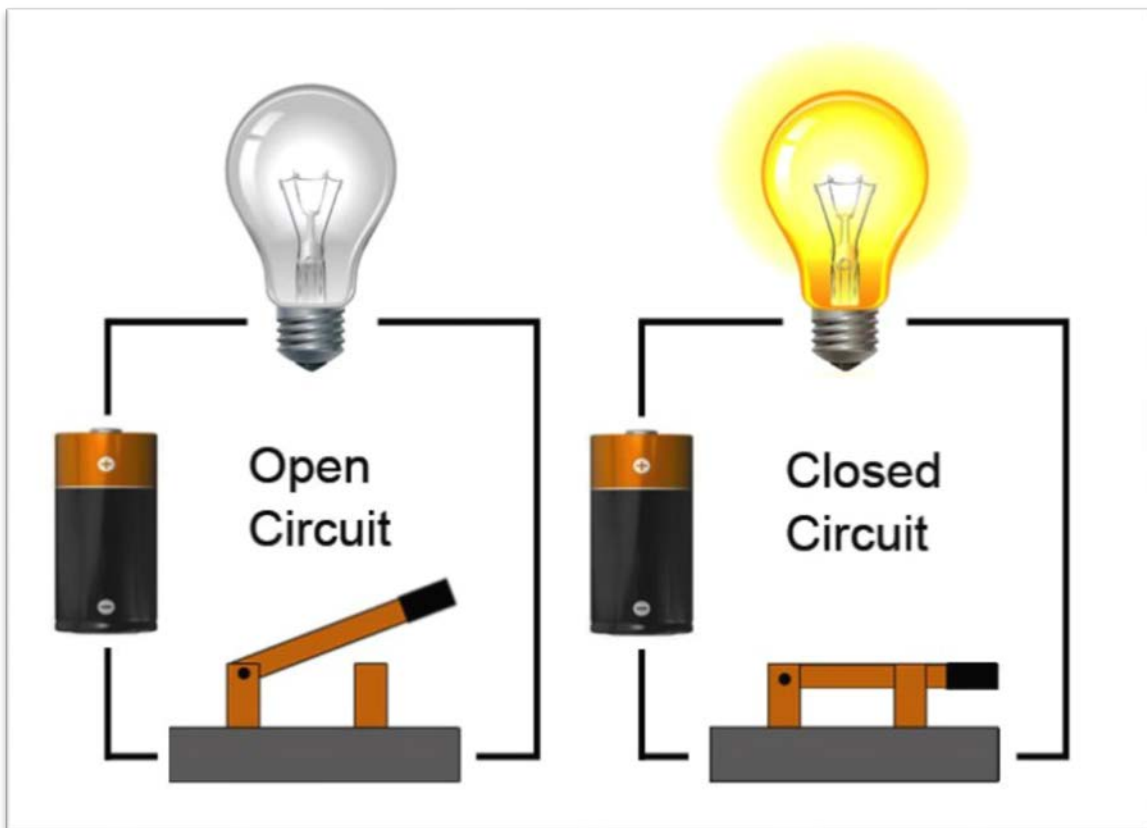


What Is A Circuit?

Before you design your electronic project, you need to know what a circuit is and how to create one properly.

An electronic circuit is a circular path of conductors by which electric current can flow. A closed circuit is like a circle because it starts and ends at the same point forming a complete loop. Furthermore, a closed circuit allows electricity to flow from the (+) power to the (-) ground uninterrupted.

In contrast, if there is any break in the flow of electricity, this is known as an open circuit. As shown below, a switch in a circuit can cause it to be open or closed depending on its position.



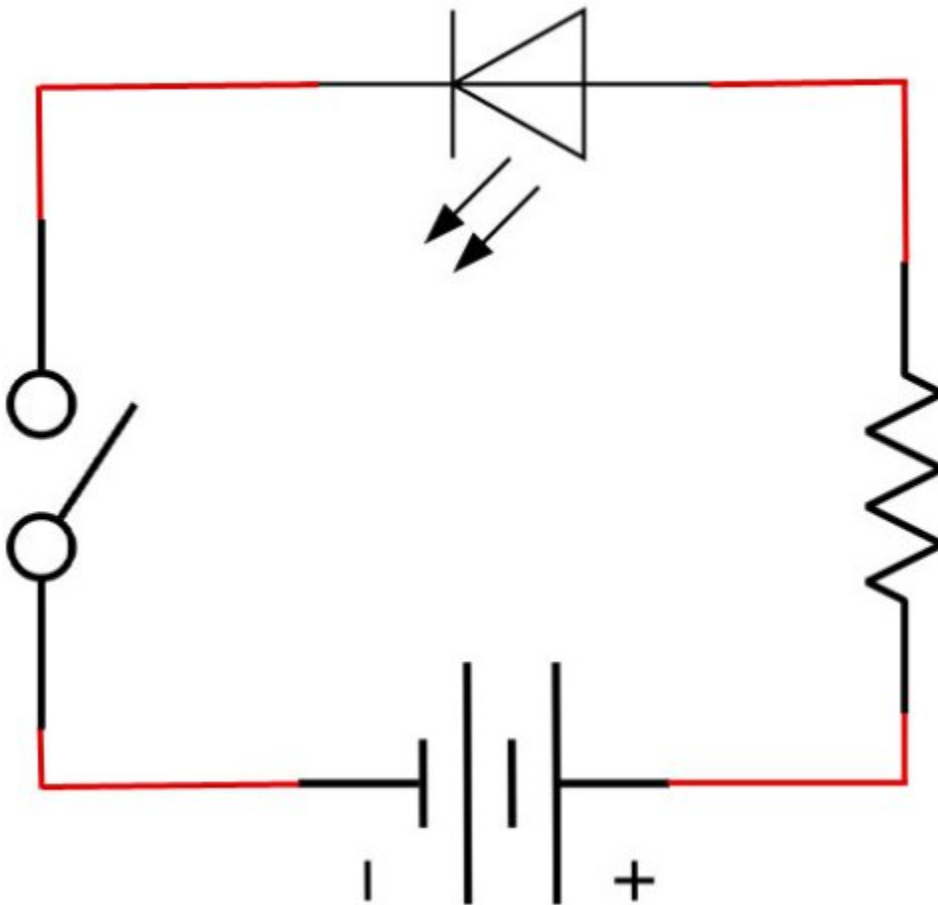
All circuits need to have three basic elements. These elements are a voltage source, conductive path and a load.

The voltage source, such as a battery, is needed in order to cause the current to flow through the circuit. In addition, there needs to be a conductive path that provides a route for the electricity to flow. Finally, a proper circuit needs a load that consumes the power. The load in the above circuit is the light bulb.

Schematic Diagrams

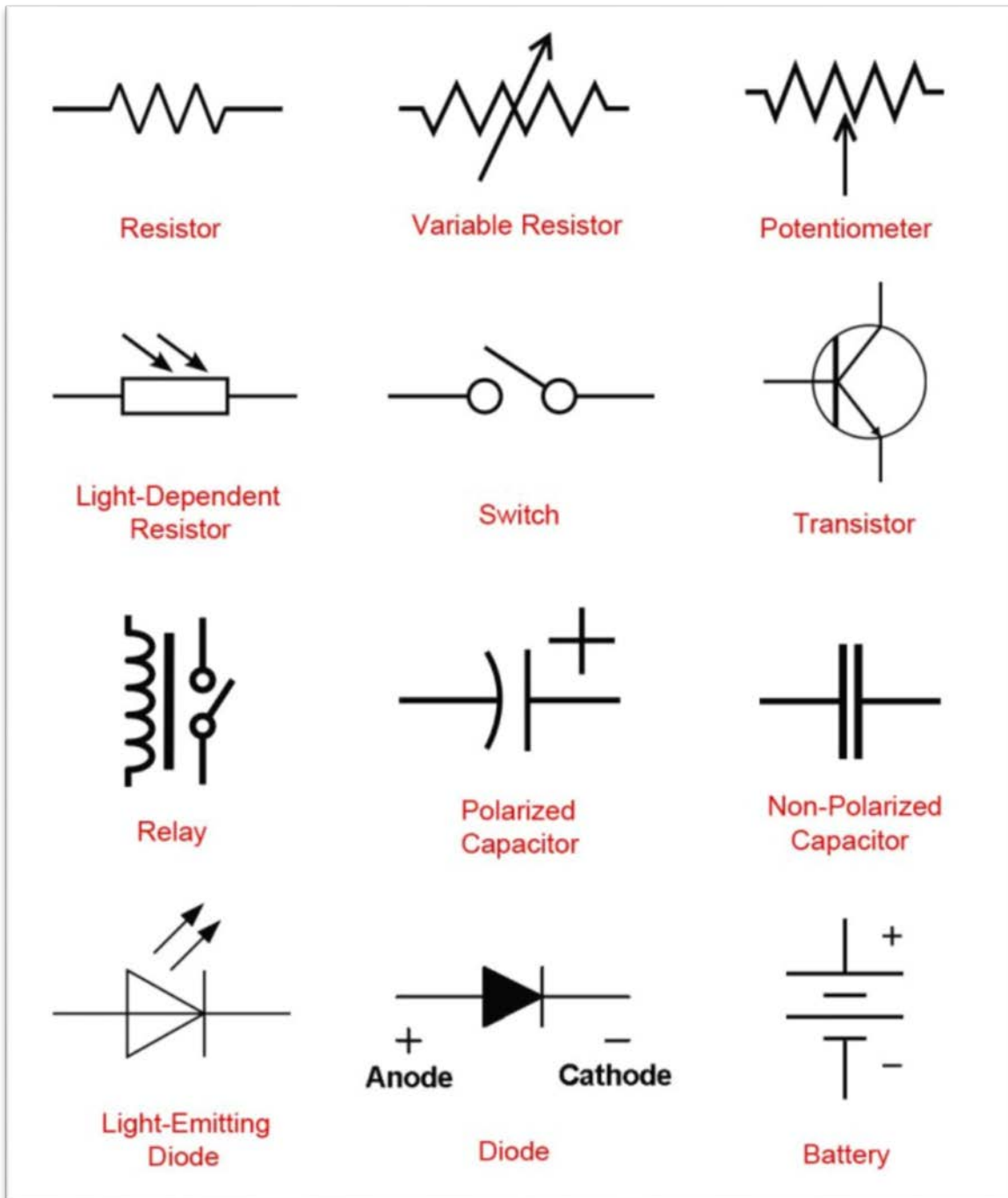
When working with circuits, you will often find something called a schematic diagram. These diagrams use symbols to illustrate what electronic components are used and where they're placed in the circuit. These symbols are graphic representations of the actual electronic components.

Below is an example of a schematic that depicts an LED circuit that is controlled by a switch. It contains symbols for an LED, resistor, battery and a switch. By following a schematic diagram, you are able to know which components to use and where to put them. These schematics are extremely helpful for beginners when first learning circuits.



Electronic Symbols

There are many types of electronic symbols and they vary slightly between countries. Below are a few of the most commonly used electronic symbols in the US.



How To Determine A Resistor Size

Resistors are commonly used in LED projects such as the ones we'll do later in this book. It's very important to know which size resistor to use or you could damage your LED. To find the proper resistor value, you need to know the voltage and the amps of your LED and battery.

A standard LED generally needs a voltage of around 2V and a current of 20mA or .02A to operate correctly. Next, you need to find out what voltage your battery is. In this example, we will be using a 9V battery. In order to determine the resistor size, we need to use a formula known as Ohm's law as shown below.

Ohm's Law – Resistance (R) = Voltage (V) / Current (I)

- Resistance is measured in Ohms (Ω)
- Voltage is measured in volts (V)
- Current is measured in amps (A)

$$R = \frac{V_{\text{Bat}} - V_{\text{LED}}}{I_{\text{LED}}}$$
$$350 = \frac{9\text{V} - 2\text{V}}{.02\text{A}}$$

Using Ohm's law, you need to subtract the LED voltage from the battery voltage. This will give you a voltage of 7 which needs to be divided by .02 amps from the LED.

This formula shows that you will need a 350 Ω resistor.

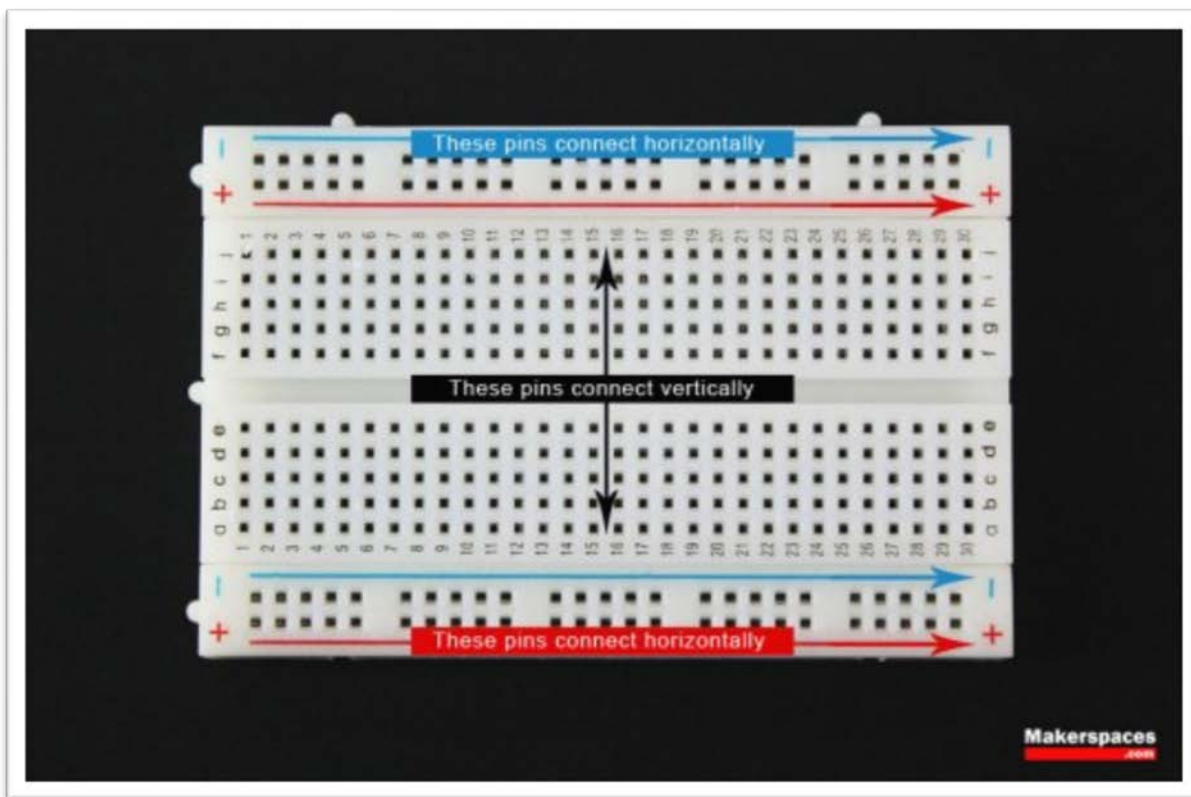
As a note, standard resistors don't come in 350 Ω but are available in 330 Ω which will work fine.

How To Use A Breadboard

Another way to create and test a circuit is to build it on a breadboard. These boards are essential for testing and prototyping circuits because no soldering is needed. Components and wires are pushed into the holes to form a temporary circuit. Because it's not permanent, you can experiment and make changes until the desired outcome is reached.

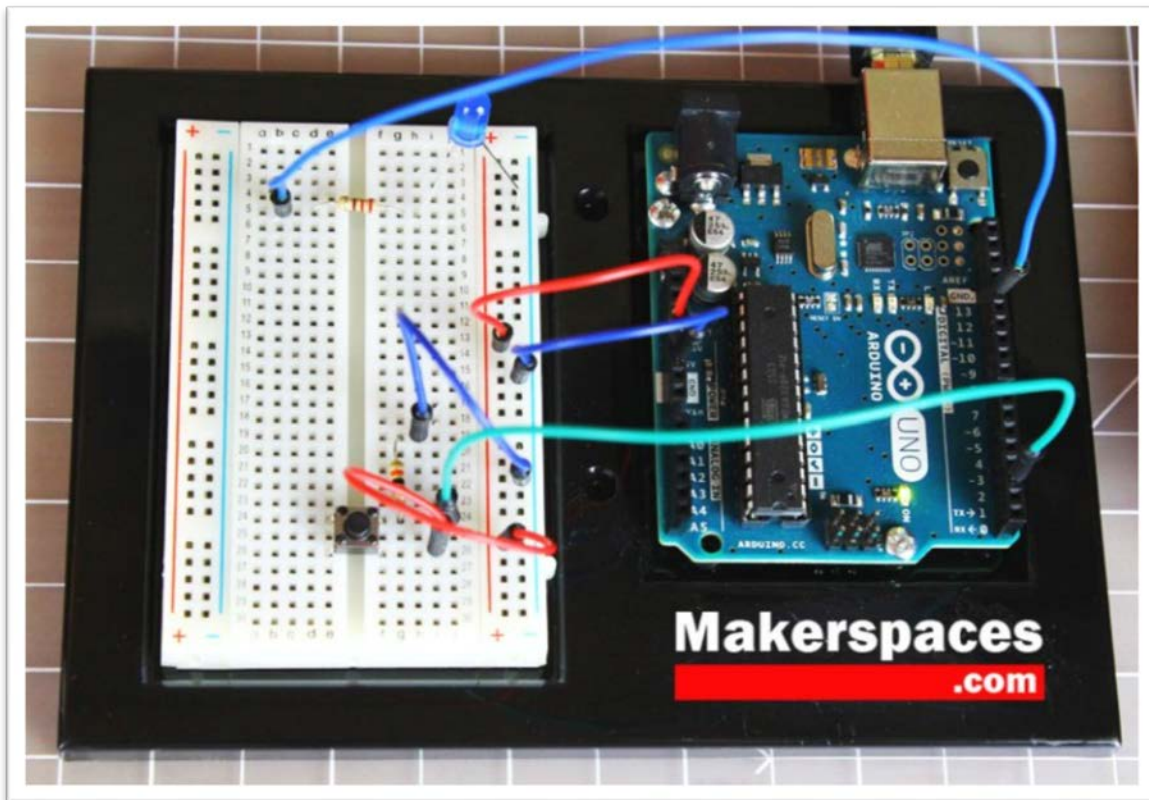
Below the holes of each row are metal clips that connect the holes to each other. The middle rows run vertically as shown while the exterior columns are connected horizontally. These exterior columns are called power rails and are used to receive and provide power to the board.

Note – There is “dip” in the middle of the breadboard between rows E & F. This dip acts as a break. The top middle section is not connected to the bottom middle section.

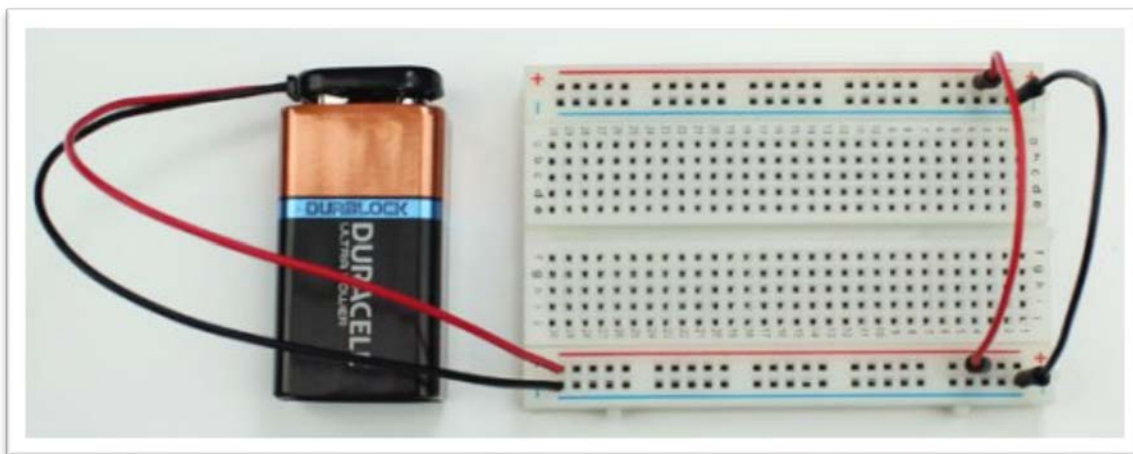


To form a circuit, insert your components and jumper wires into the breadboard. Later, you will create a few breadboard circuits which will help you truly understand how they work.

Here is a visual of what a completed Arduino circuit looks like when connected to a breadboard. The colors of the wires don't matter and you can use any color or length.



Breadboards will need to have power supplied to them and this can be done in a few ways. One of the easiest ways is to plug the wires from a battery holder into the power rails. This will supply voltage to the rail it's plugged into only.



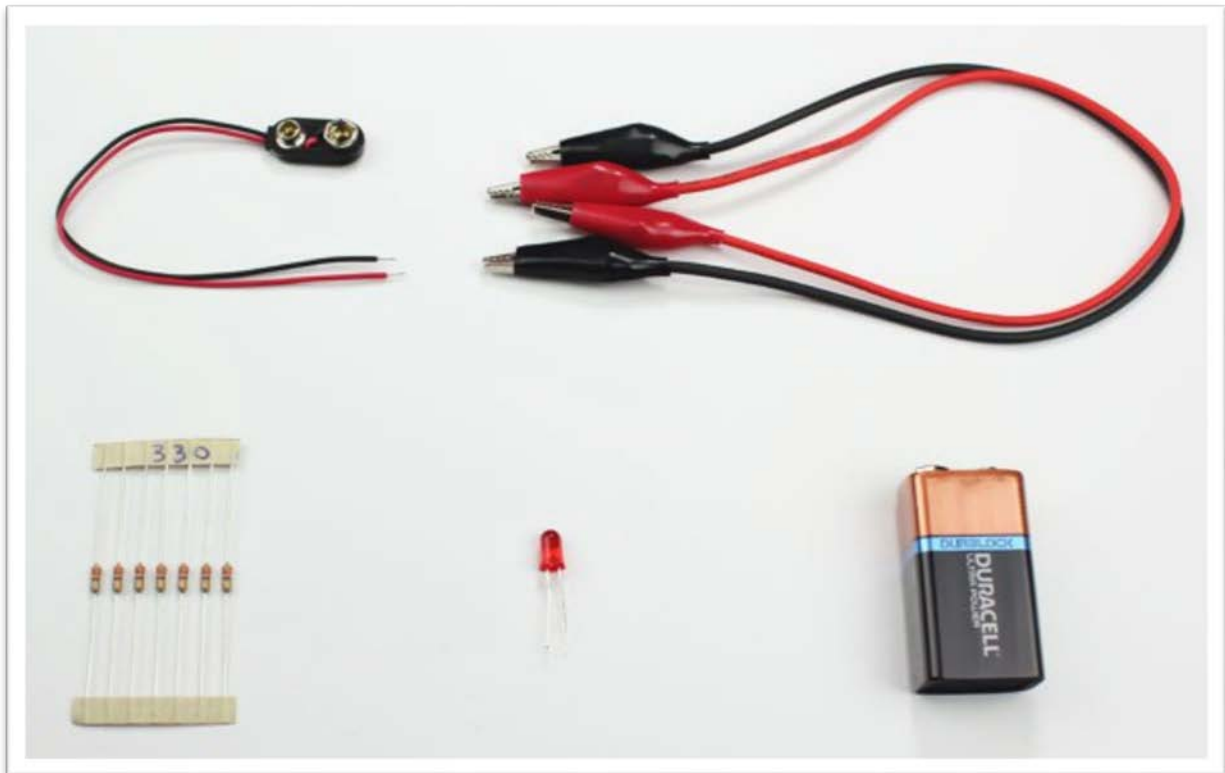
To power both rails, you will need to use a jumper wire from the (+) and (-) to the rail on the opposite side. Now both sides of the breadboard are powered with 9V.

Practice Project #1

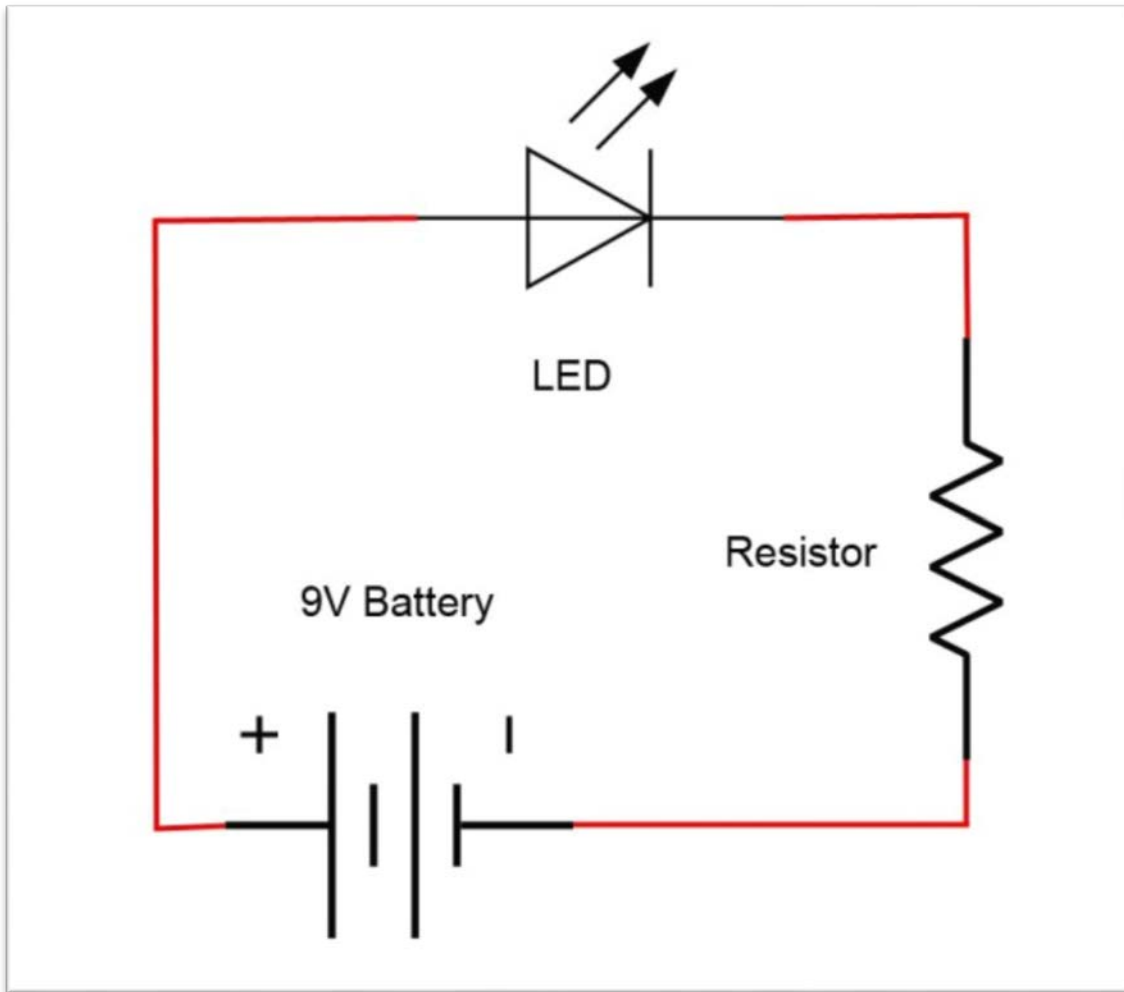
In our first project, we'll be using test leads to create a very simple LED circuit. In our next project, we will create this same circuit but with a breadboard.

Parts Needed:

- [9V Battery](#)
- [Battery Snap-on Connector](#)
- [Test Leads w/ Alligator Clips](#)
- [330 Ohm Resistor](#)
- [LED – Basic Red 5mm](#)

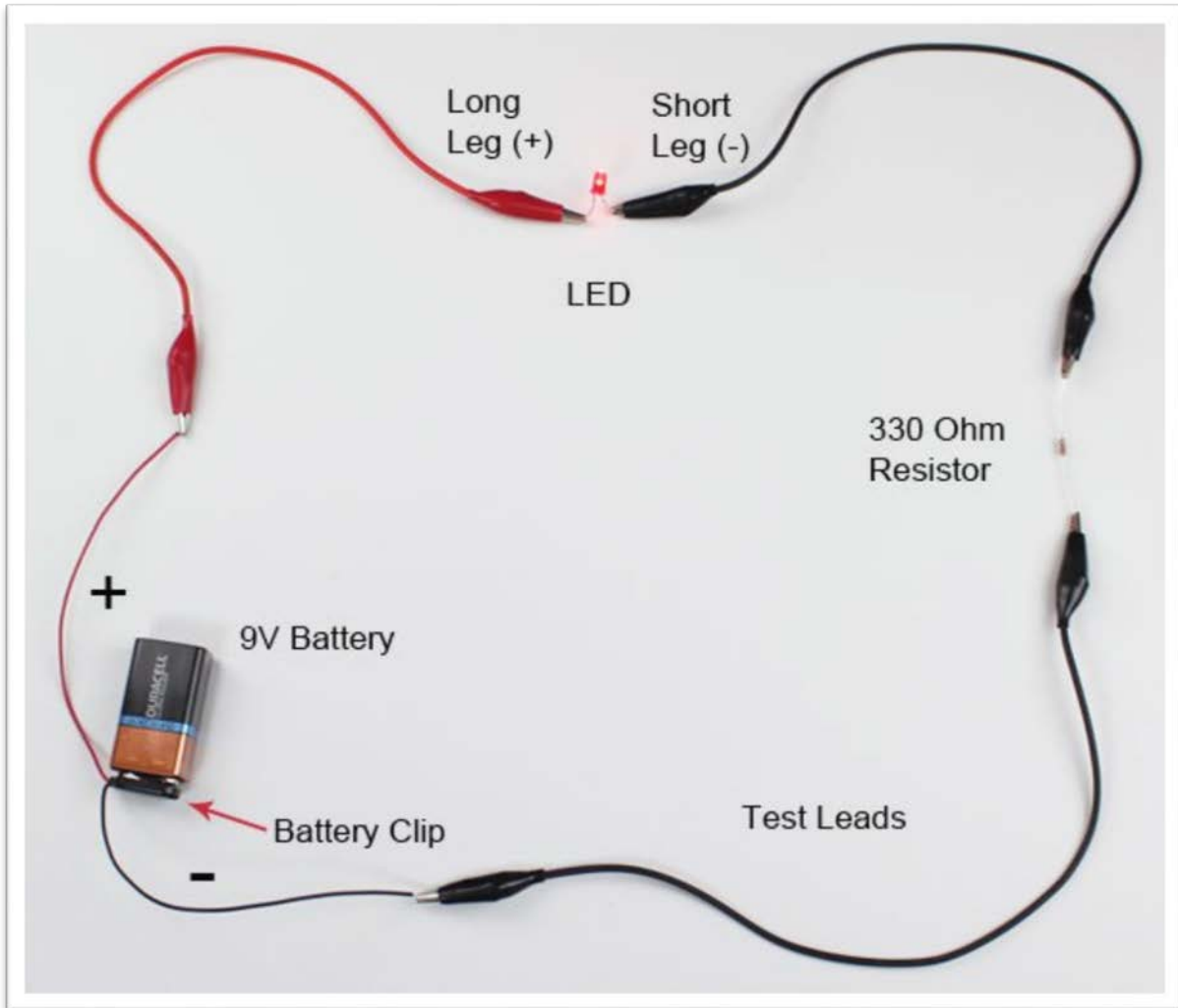


Schematic Diagram



Project Steps

1. Attach the battery clip to the top of the 9V battery.
2. Red wire from the battery clip is connected to one alligator clip on the red test lead.
3. The other end of the red test lead is connected to the long leg (+) of the LED.
4. Connect one alligator clip from black test lead to the short leg (-) of the LED.
5. The other end of the black test lead is clipped to one leg of the 330 Ω resistor.
6. Clip one side of the other black test lead to the other leg of the 330 Ω resistor.
7. The opposite end of the black test lead is connected to the black battery wire.



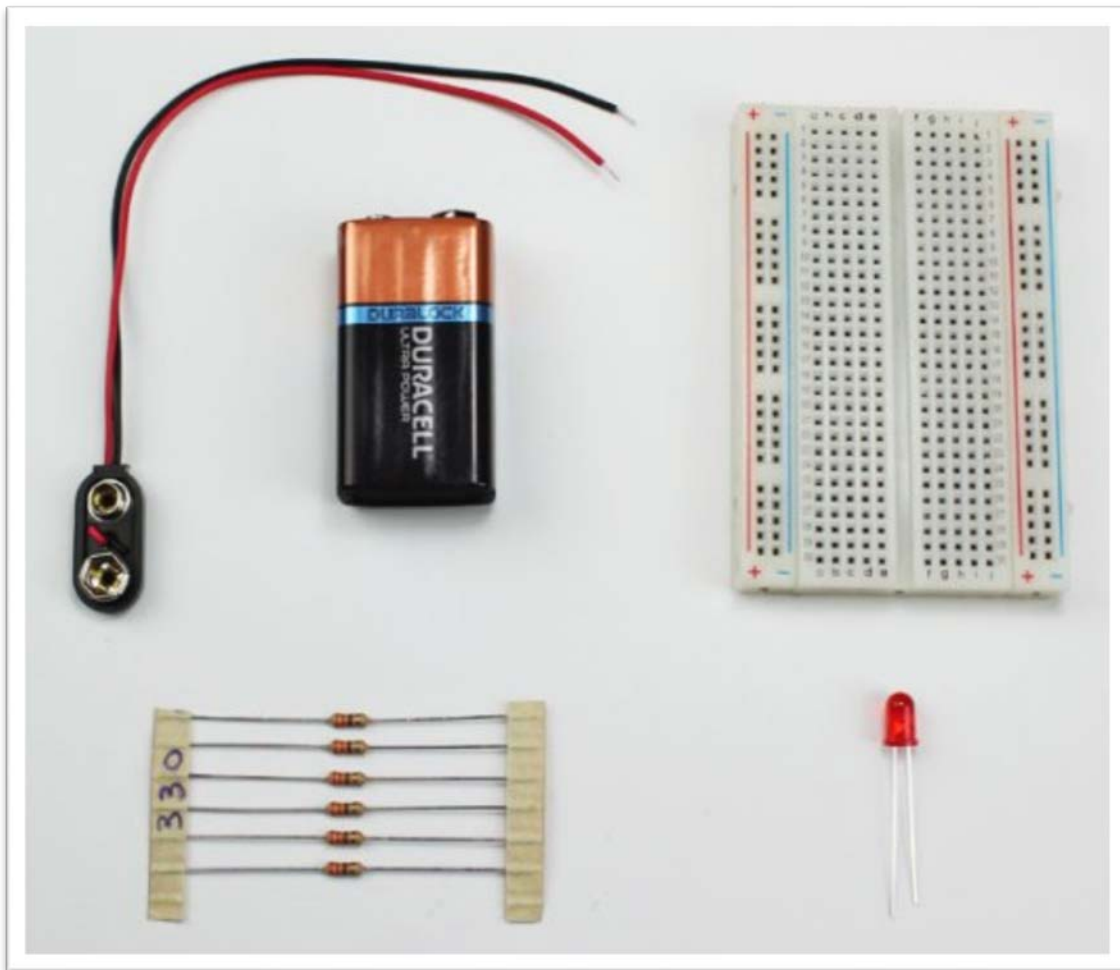
IMPORTANT – Never connect an LED directly to a 9V battery without a resistor in the circuit. Doing so will damage or destroy the LED.

Practice Project #2

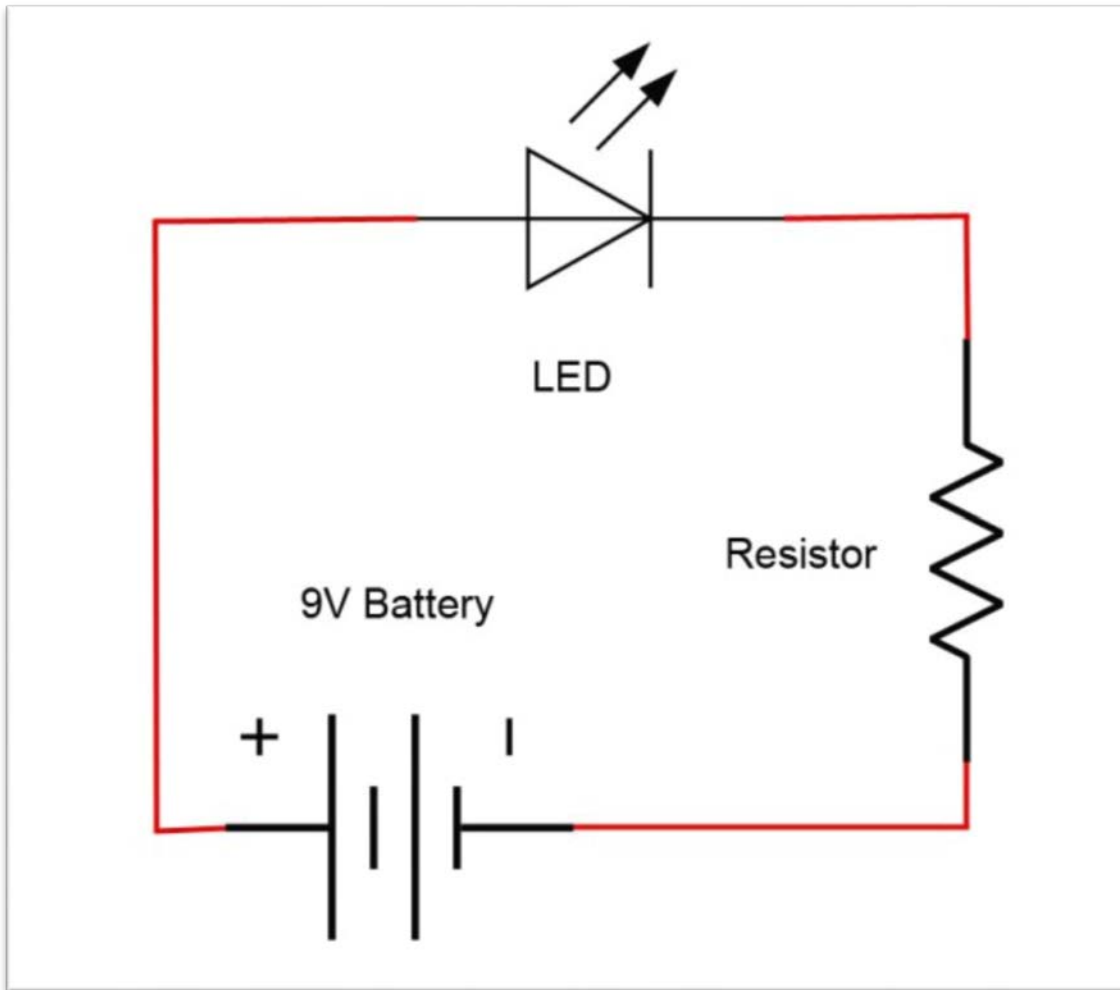
Now we're going to learn how to create a circuit on a breadboard. This circuit is the exact same one we did earlier but without the test leads.

Parts Needed:

- [9V Battery](#)
- [Battery Snap-on Connector](#)
- [330 Ohm Resistor](#)
- [LED – Basic Red 5mm](#)
- [Breadboard- Half Size](#)

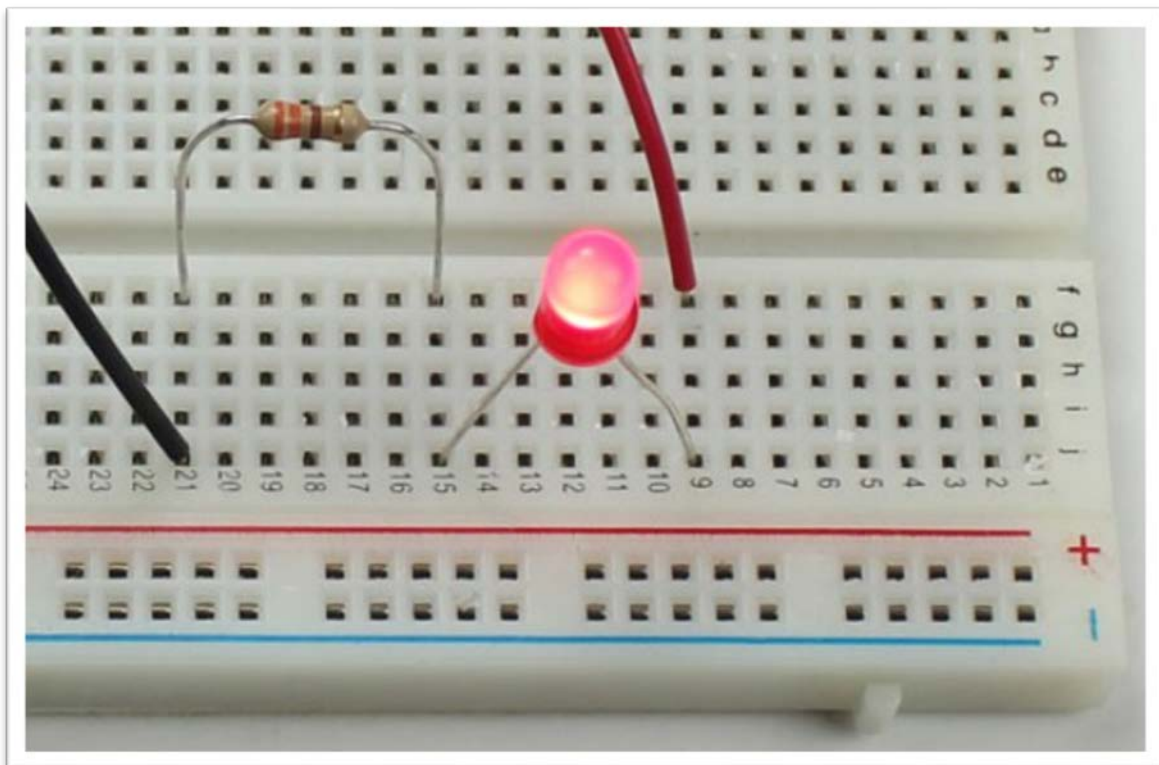
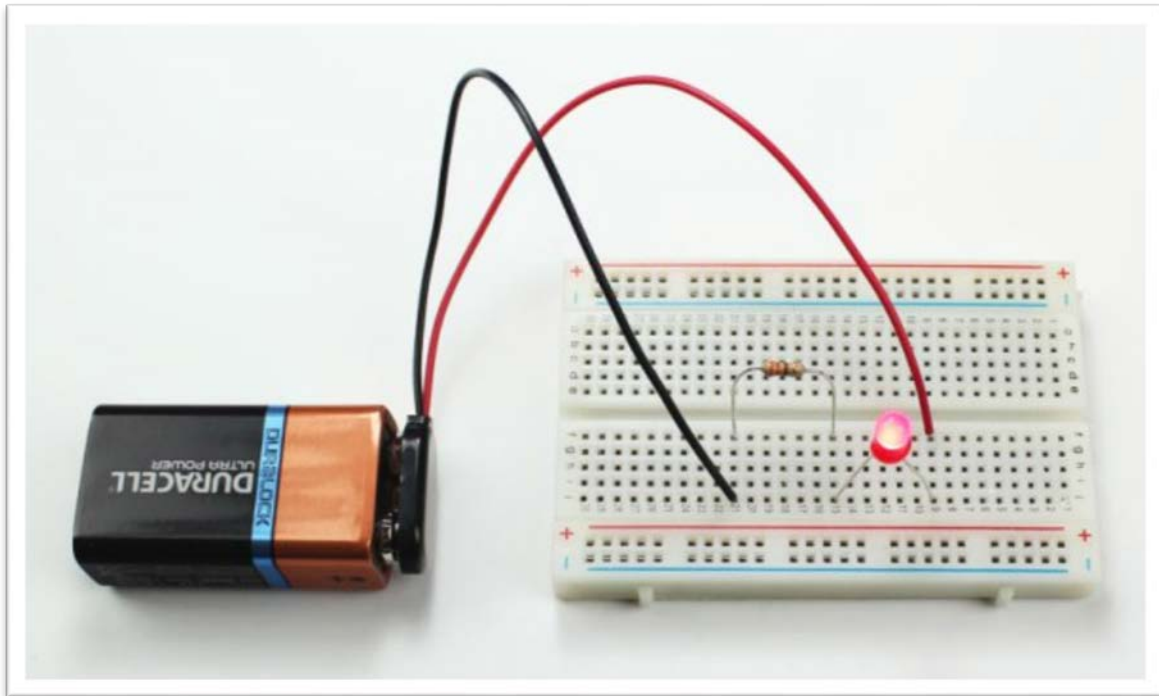


Schematic Diagram

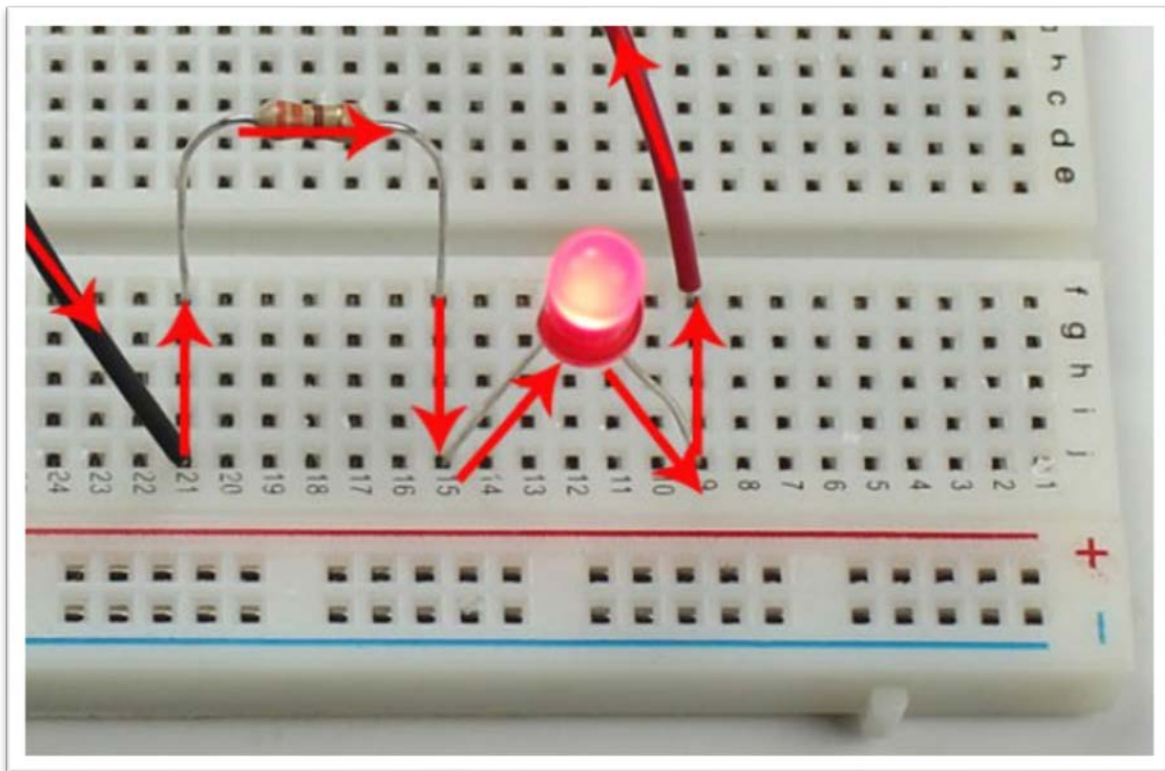


Project Steps

1. Attach the battery clip to the top of the 9V battery.
2. Place the red wire from the battery clip into F9 of the breadboard.
3. Insert the black wire from the battery clip into J21 of the breadboard.
4. Bend the legs of the 330 Ω resistor and place one leg into F21.
5. Place the other leg of the resistor into F15.
6. Insert the short leg of the LED into J15 and the long leg into J9.



The red arrows in the image below help to show how electricity is flowing in this circuit. All components are connected to each other in a circle just like when we used the test leads.



IMPORTANT – Never connect an LED directly to a 9V battery without a resistor in the circuit. Doing so will damage or destroy the LED.

3

Arduino Hardware Basics

Arduino Hardware Basics



What Is Arduino?

Arduino is an open source programmable circuit board that can be integrated into a wide variety of makerspace projects both simple and complex. This board contains a microcontroller which is able to be programmed to sense and control objects in the physical world. By responding to sensors and inputs, the Arduino is able to interact with a large array of outputs such as LEDs, motors and displays. Because of its flexibility and low cost, Arduino has become a very popular choice for makers and makerspaces looking to create interactive hardware projects.

Arduino was introduced back in 2005 in Italy by Massimo Banzi as a way for non-engineers to have access to a low cost, simple tool for creating hardware projects. Since the board is open-source, it is released under a Creative Commons license which allows anyone to produce their

own board. If you search the web, you will find there are hundreds of Arduino compatible clones and variations available but the only official boards have Arduino in its name.

In the next section, we're going to discuss a few of the Arduino boards available and how they differ from each other.



Types of Arduino Boards

Arduino is a great platform for prototyping projects and inventions but can be confusing when having to choose the right board. If you're brand new to this, you might have always thought that there was just one "Arduino" board and that's it. In reality, there are many variations of the official Arduino boards and then there are hundreds more from competitors who offer clones. But don't worry, we're going to show you which one to start with later on in this tutorial.

Below are a few examples of the different types of Arduino boards out there. The boards with the name Arduino on them are the official boards but there are also a lot of really great clones on the market as well. One of the best reasons to buy a clone is the fact they are generally less expensive than their official counterpart. [Adafruit](#) and [Sparkfun](#) for example, sell variations of the Arduino boards which cost less but still have the same quality of the originals. Be careful when buying boards from companies you don't know as the quality can vary.

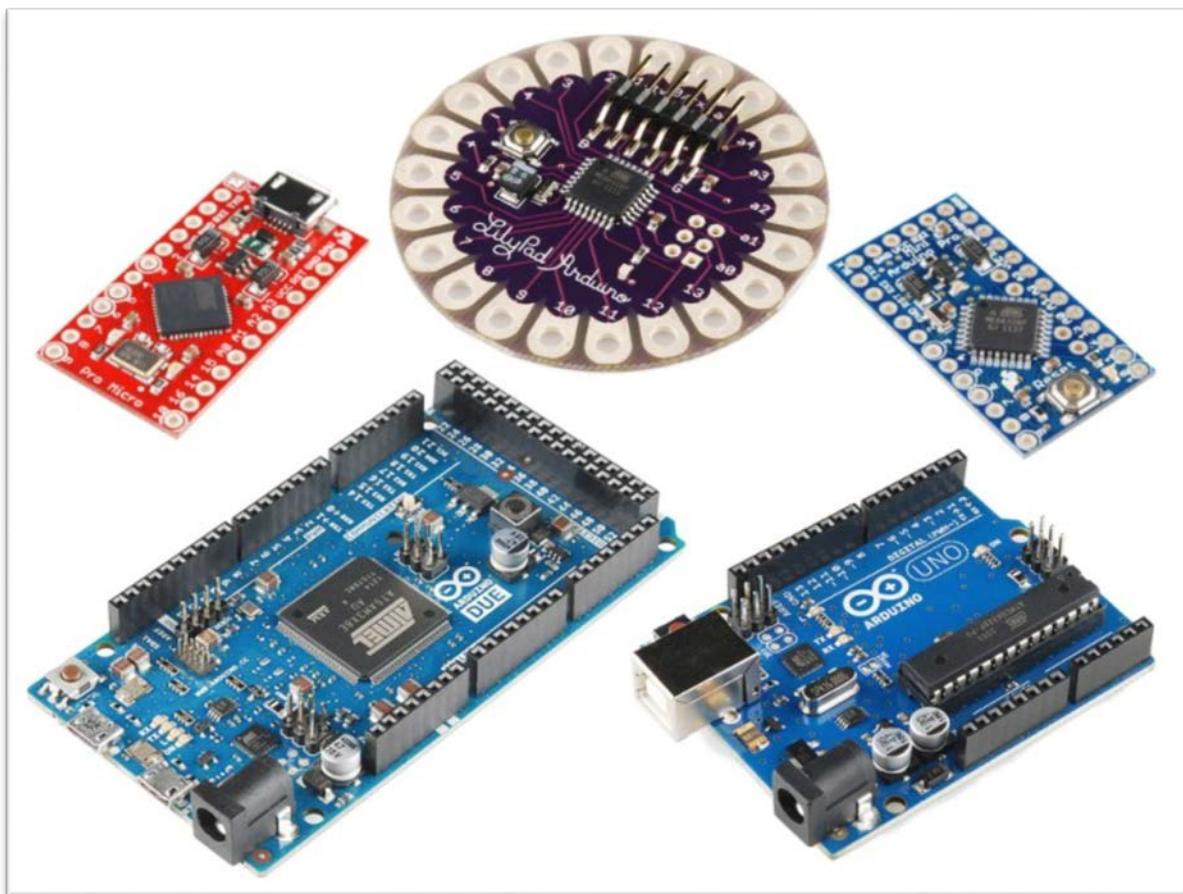
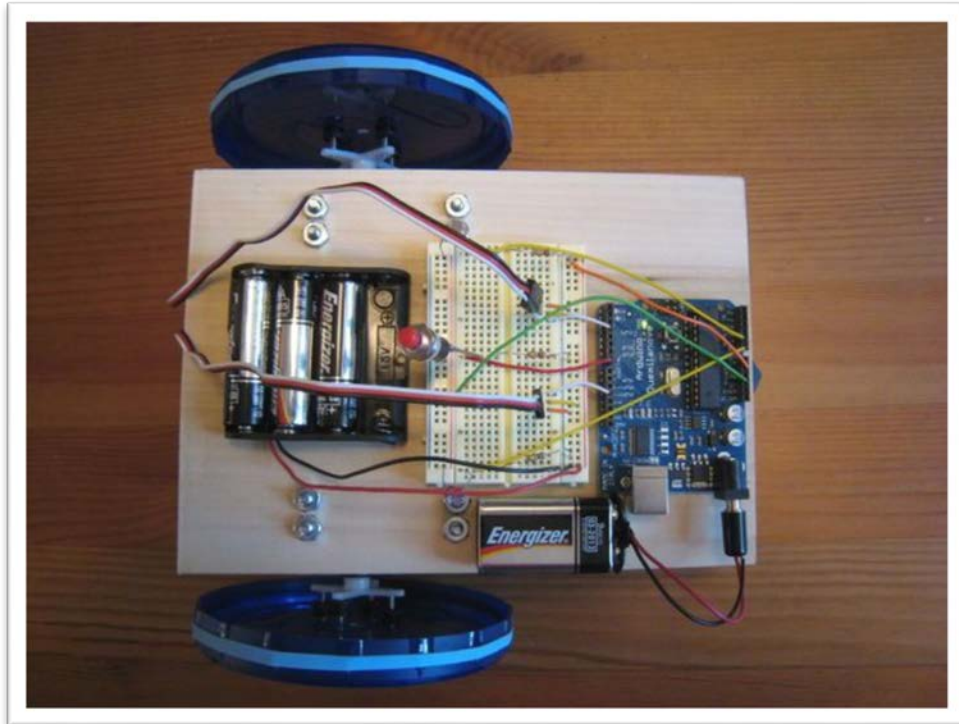


Image credit – Sparkfun.com

Another factor to consider when choosing a board is the type of project you are looking to do. For example, if you want to create a wearable electronic project, you might want to consider the LilyPad board from Sparkfun. The LilyPad is designed to be easily sewn into e-textiles and wearable projects. If your project has a small form factor, you might want to use the Arduino Pro Mini which has a very small footprint compared to other boards.

What Can You Do With Arduino?

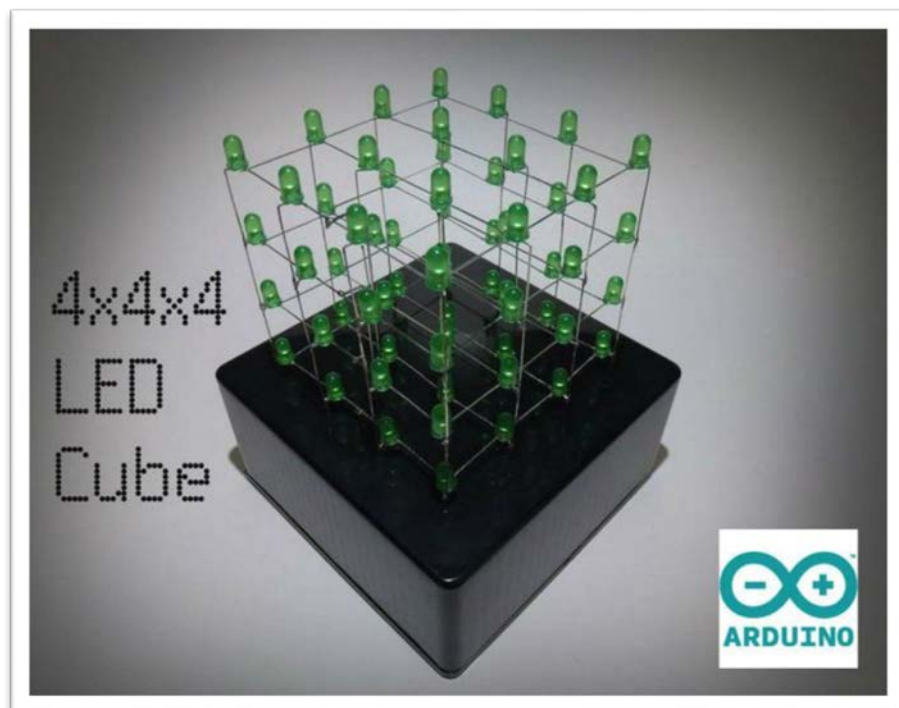
You may be wondering what an Arduino board can do besides blink an LED. Below are some example projects which help to showcase how truly amazing this board is and the capabilities of it. If you're looking for more project ideas, check out sites such as [Instructables](https://www.instructables.com/) or [Make Magazine](https://www.makeamagazine.com/) which are loaded with helpful tutorials.



[Arduino Light Following Robot](https://www.instructables.com/) – Instructables



[Arduino Drone That Follows You](#) – Instructables

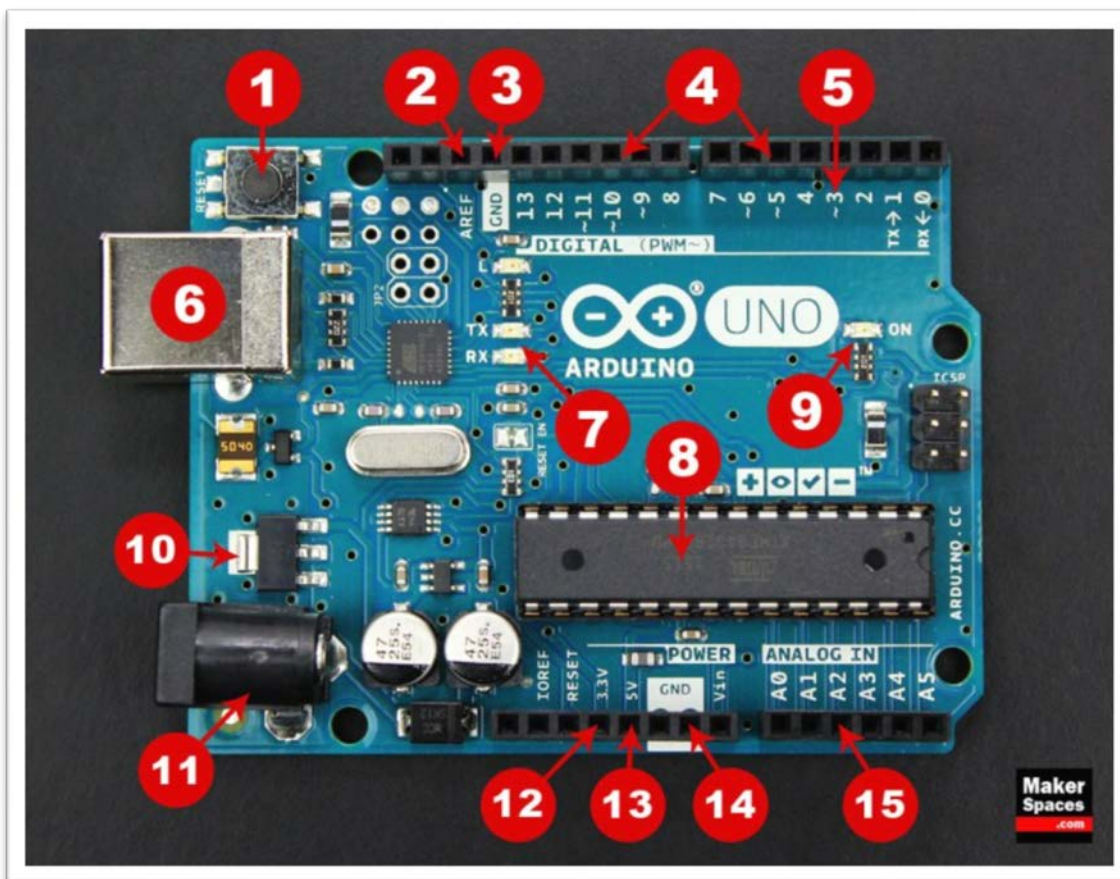


[LED Cube w/ Arduino Uno](#) – Instructables

Arduino Uno Breakdown

One of the most popular Arduino boards out there is the Arduino Uno. While it was not actually the first board to be released, it remains to be the most actively used and most widely documented on the market. Because of its extreme popularity, the Arduino Uno has a ton of project tutorials and forums around the web that can help you get started or out of a jam. We're big fans of the Uno because of its amazing features and ease of use.

All of the practice projects in this book will use an Arduino Uno. Below is a breakdown of the individual components that make up a board. Even though there are a lot of parts to the board, you only interact with a few of them.



Board Breakdown

1. **Reset Button** – This will restart any code that is loaded to the Arduino board
2. **AREF** – Stands for “Analog Reference” and is used to set an external reference voltage
3. **Ground Pin** – There are a few ground pins on the Arduino and they all work the same
4. **Digital Input/Output** – Pins 0-13 can be used for digital input or output
5. **PWM** – The pins marked with the (~) symbol can simulate analog output
6. **USB Connection** – Used for powering up your Arduino and uploading sketches
7. **TX/RX** – Transmit and receive data indication LEDs
8. **ATmega Microcontroller** – This is the brains and is where the programs are stored
9. **Power LED Indicator** – This LED lights up anytime the board is plugged in a power source
10. **Voltage Regulator** – This controls the amount of voltage going into the Arduino board
11. **DC Power Barrel Jack** – This is used for powering your Arduino with a power supply
12. **3.3V Pin** – This pin supplies 3.3 volts of power to your projects
13. **5V Pin** – This pin supplies 5 volts of power to your projects
14. **Ground Pins** – There are a few ground pins on the Arduino and they all work the same
15. **Analog Pins** – These pins can read the signal from an analog sensor and convert it to digital

Powering The Board

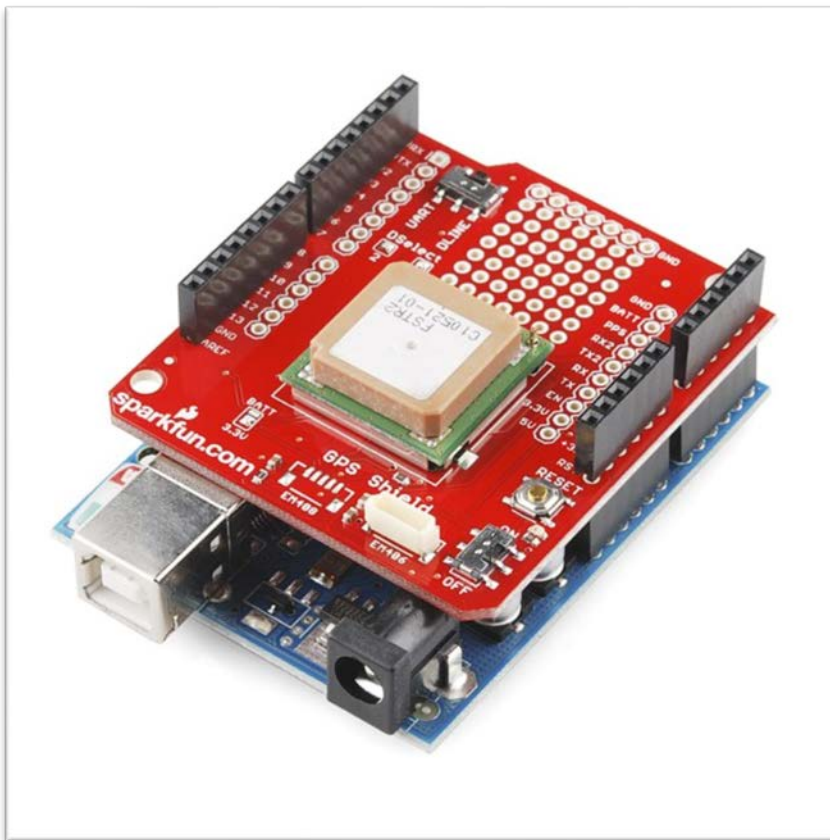
The Arduino Uno needs a power source in order for it to operate and can be powered in a variety of ways. You can do what most people do and connect the board directly to your computer via a USB cable. If you want your project to be mobile, consider using a 9V battery pack to give it juice. The last method would be to use a 9V AC power supply.



Arduino Shields

If you want to add a very specific functionality to your Arduino, you will need to use a shield. Arduino shields plug into the top of the Arduino board and can add capabilities such as WiFi, Bluetooth, GPS and much more. There are literally hundreds of shields to choose from and here are a few examples.

- [WiFi Shield](#)
- [LCD Shield](#)
- [GPS Logger Shield](#)
- [MP3 Music Maker Shield](#)
- [Ethernet Shield](#)
- [Motor/Stepper/Servo Shield](#)

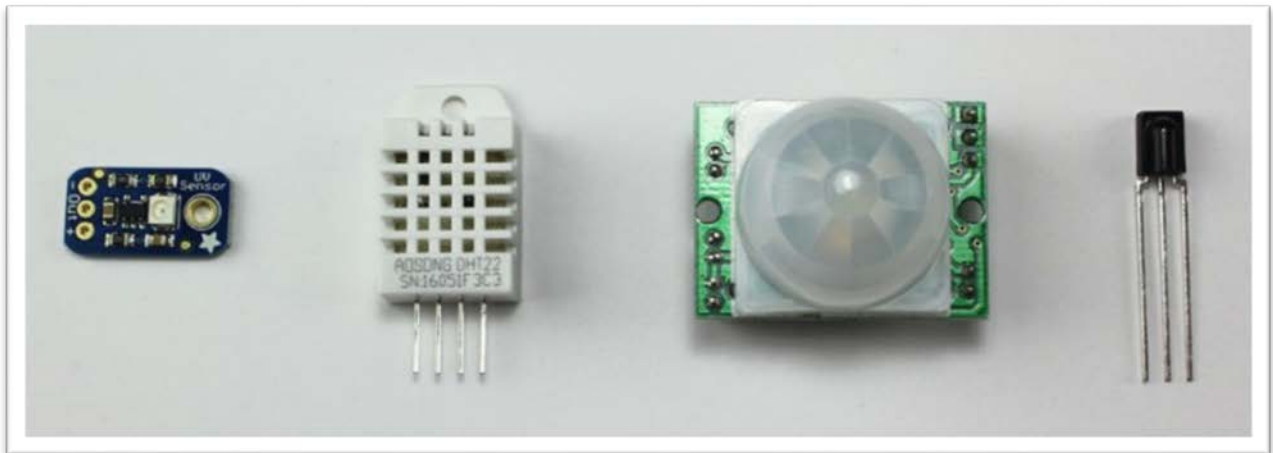


GPS Shield Plugged into Arduino Uno – Sparkfun.com

Arduino Sensors

If you want your Arduino to sense the world around it, you will need to add a sensor. There are a wide range of sensors to choose from and they each have a specific purpose. Below you will find some of the commonly used sensors in projects.

- [Distance Ranging Sensor](#)
- [PIR Motion Sensor](#)
- [Light Sensor](#)
- [Degree of Flex Sensor](#)
- [Pressure Sensor](#)
- [Proximity Sensor](#)
- [Acceleration Sensor](#)
- [Sound Detecting Sensor](#)
- [RGB and Gesture Sensor](#)
- [Humidity and Temperature Sensor](#)



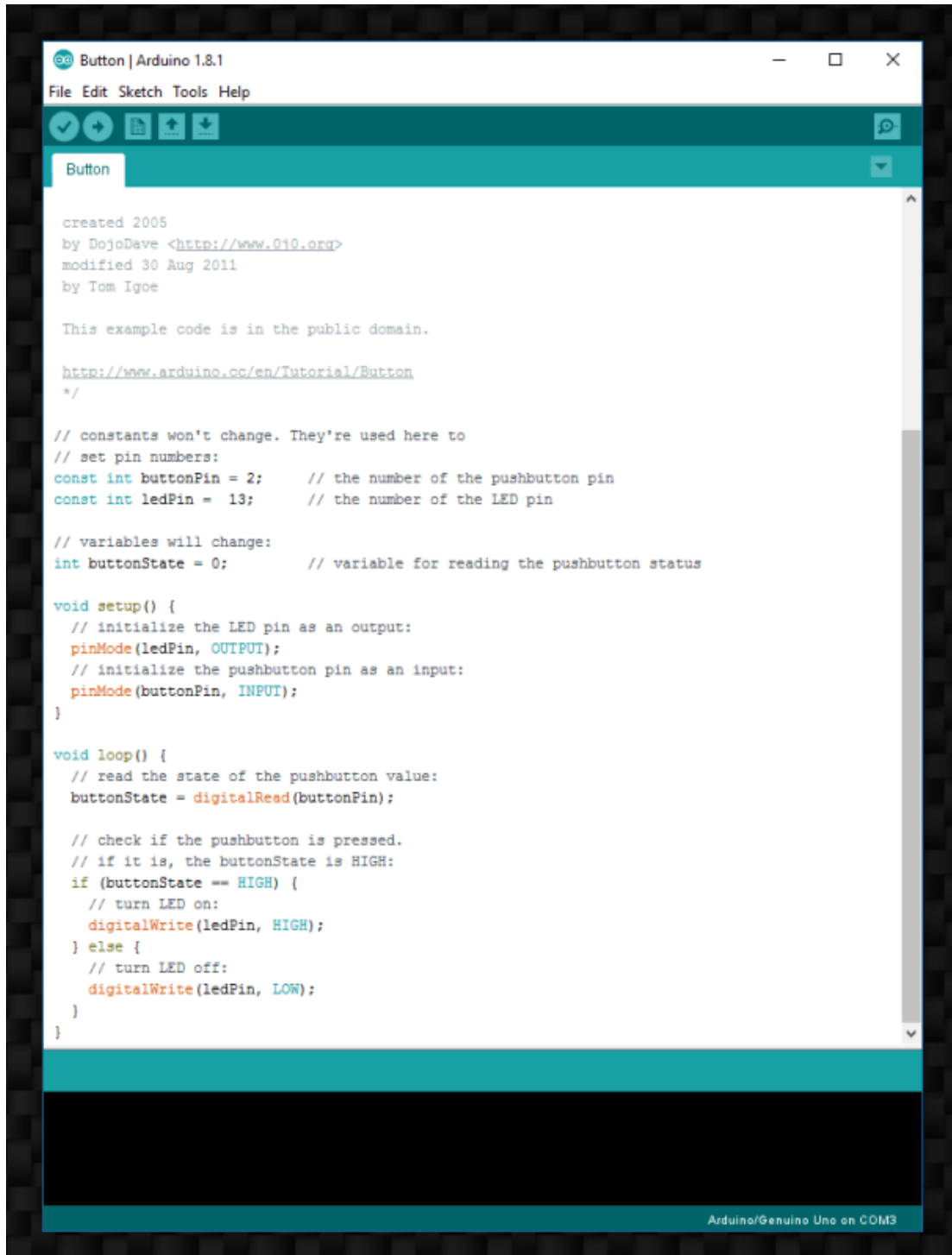
Examples of Arduino Sensors

4

Arduino Software Basics

Arduino Software Basics

This chapter will focus on the software portion of Arduino. The easiest way to learn how the software works is if you download it and look around. Don't worry, it's free!



Arduino Programming Software (IDE)

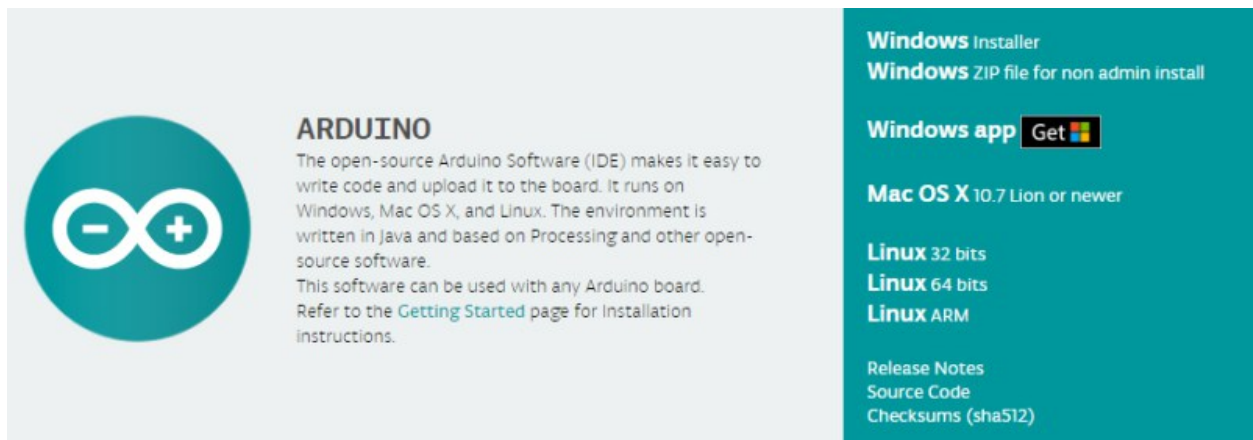
The image on the previous page shows the Arduino software which is called the Integrated Development Environment or IDE. This is the place where you create the programs known as sketches. These sketches are then uploaded to the Arduino which tells the board what to do.

Download The Arduino IDE

The Arduino IDE is a free program which can be downloaded at the following link:

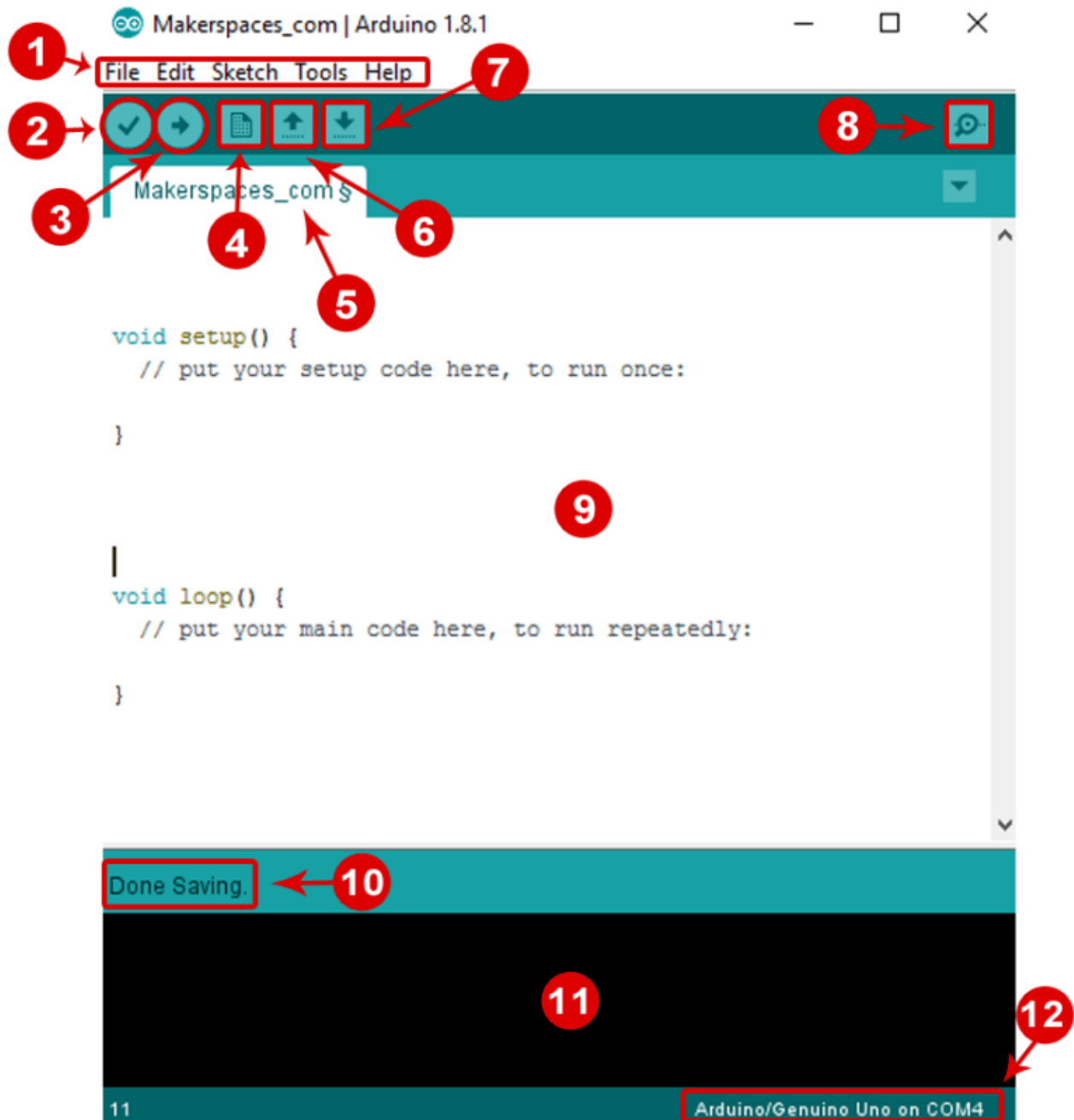
<https://www.arduino.cc/en/Main/Software>

Select the operating system you have in order to begin the download process.



Arduino Software Breakdown

Once the software has been installed on your computer, go ahead and open it up. Below is a breakdown of the software interface.



1. **Menu Bar:** Gives you access to the tools needed for creating and saving Arduino sketches.
2. **Verify Button:** Compiles your code and checks for errors in spelling or syntax.
3. **Upload Button:** Sends the code to the board that's connected such as Arduino Uno in this case. Lights on the board will blink rapidly when uploading.
4. **New Sketch:** Opens up a new window containing a blank sketch.
5. **Sketch Name:** When the sketch is saved, the name of the sketch is displayed here.
6. **Open Existing Sketch:** Allows you to open a saved sketch or one from the stored examples.
7. **Save Sketch:** This saves the sketch you currently have open.
8. **Serial Monitor:** When the board is connected, this will display the serial information of your Arduino
9. **Code Area:** This area is where you compose the code of the sketch that tells the board what to do.
10. **Message Area:** This area tells you the status on saving, code compiling, errors and more.
11. **Text Console:** Shows the details of error messages, size of the program that was compiled and additional info.
12. **Board and Serial Port:** Tells you what board is being used and what serial port it's connected to.

Every Arduino sketch has two main parts to the program which are shown below. You can see this in the code area (#9) of the interface.

void setup() – Sets things up that have to be done once and then don't happen again.

void loop() – Contains the instructions that get repeated over and over until the board is turned off.

5

Your First Arduino Project

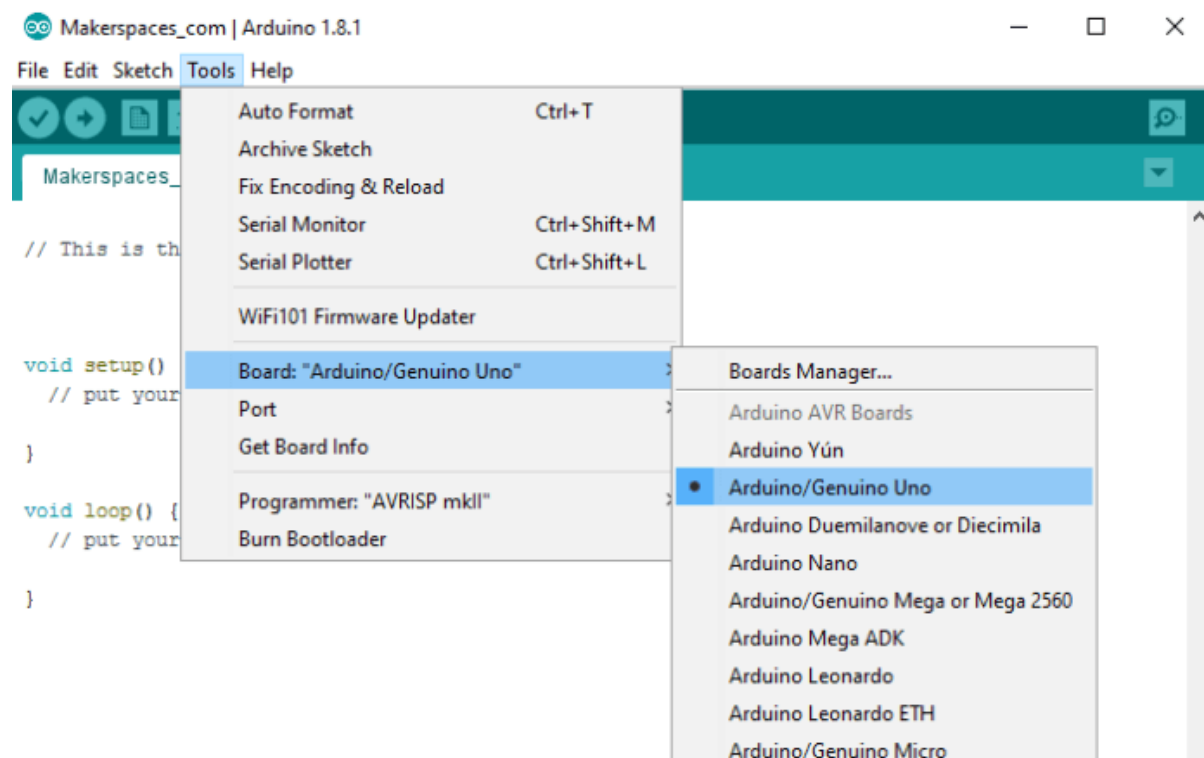
Your First Arduino Project

Now it's time to take what you learned so far and create your first Arduino project. Once you finish this project you will start to understand how everything works together.

Connect Your Arduino Uno

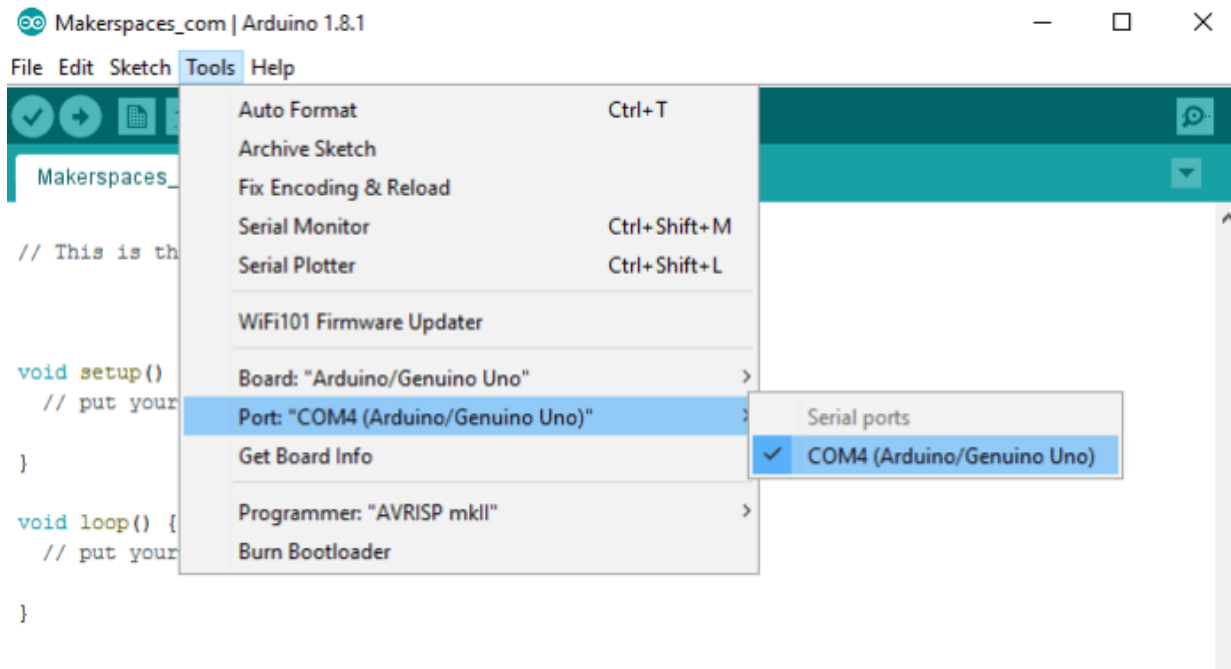
At this point you are ready to connect your Arduino to your computer. Plug one end of the USB cable to the Arduino Uno and then the other end of the USB to your computer's USB port.

Once the board is connected, you will need to open the IDE and go to **Tools** then **Board** then finally select **Arduino Uno**.



Next, you have to tell the Arduino which port you are using on your computer.

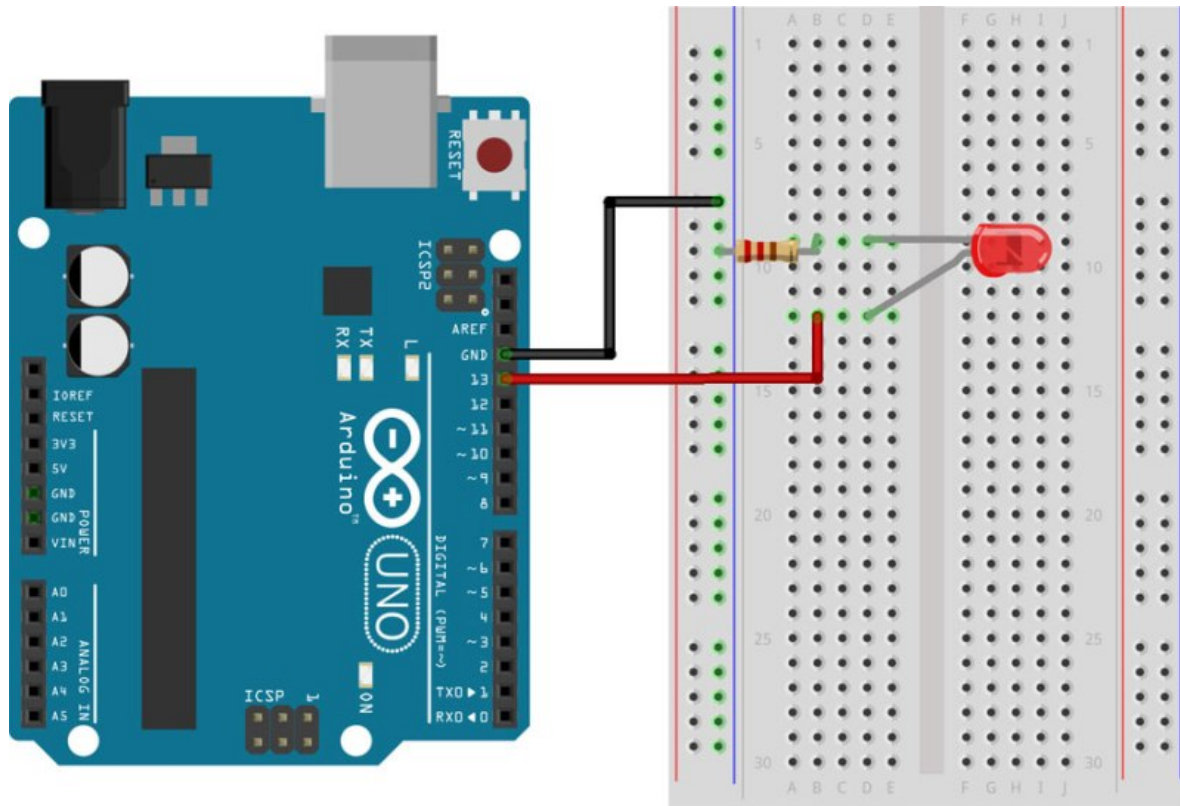
To select the port, go to **Tools** then **Port** then select the COM port that says **Arduino**.



Now that the board is connected, we're ready for your first Arduino project.

Arduino Project 1: Blink an LED

In this example, we are going to tell your Arduino board to blink an LED.



Required Parts

- [Arduino Uno Board](#)
- [Breadboard](#) – half size
- [Jumper Wires](#)
- [USB Cable](#)
- [LED \(5mm\)](#)
- [220 Ohm Resistor](#)

Connect The Parts

You can build your Arduino circuit by looking at the breadboard image or by using the written description below. In the written description, we will use a letter/number combo that refers to the location of the component. If we mention H19 for example, that refers to column H, row 19 on the breadboard.

Step 1 – Insert black jumper wire into the GND (Ground) pin on the Arduino and then in the GND rail of the breadboard row 15

Step 2 – Insert red jumper wire into pin 13 on the Arduino and then the other end into F7 on the breadboard

Step 3 – Place the LONG leg of the LED into H7

Step 4 – Place the SHORT leg of the LED into H4

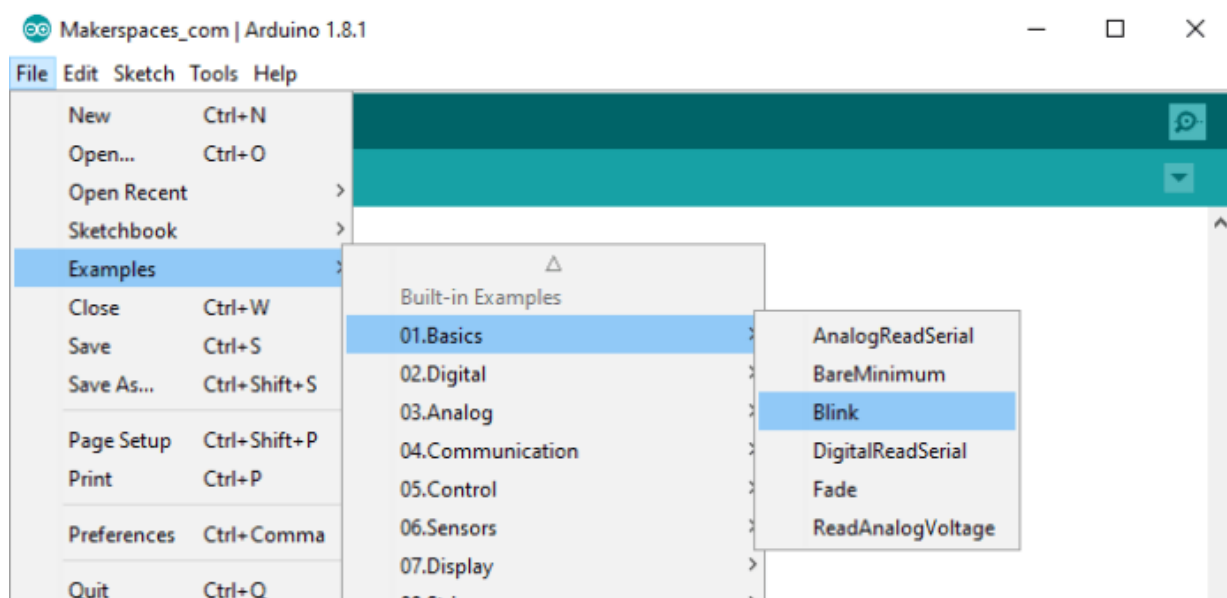
Step 5 – Bend both legs of a 220 Ohm resistor and place one leg in the GND rail around row 4 and other leg in I4

Step 6 – Connect the Arduino Uno to your computer via USB cable

Upload The Blink Sketch

Now it's time to upload the sketch (program) to the Arduino and tell it what to do. In the IDE, there are built-in example sketches that you can use. We're going to use the blink sketch.

To open the blink sketch, you will need to go to **File > Examples > Basics > Blink**



Now you should see a fully coded blink sketch that looks like the image below.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino model, check
  the Technical Specs of your board at https://www.arduino.cc/en/Main/Products

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald

  modified 2 Sep 2016
  by Arturo Guadalupi

  modified 8 Sep 2016
  by Colby Newman
*/

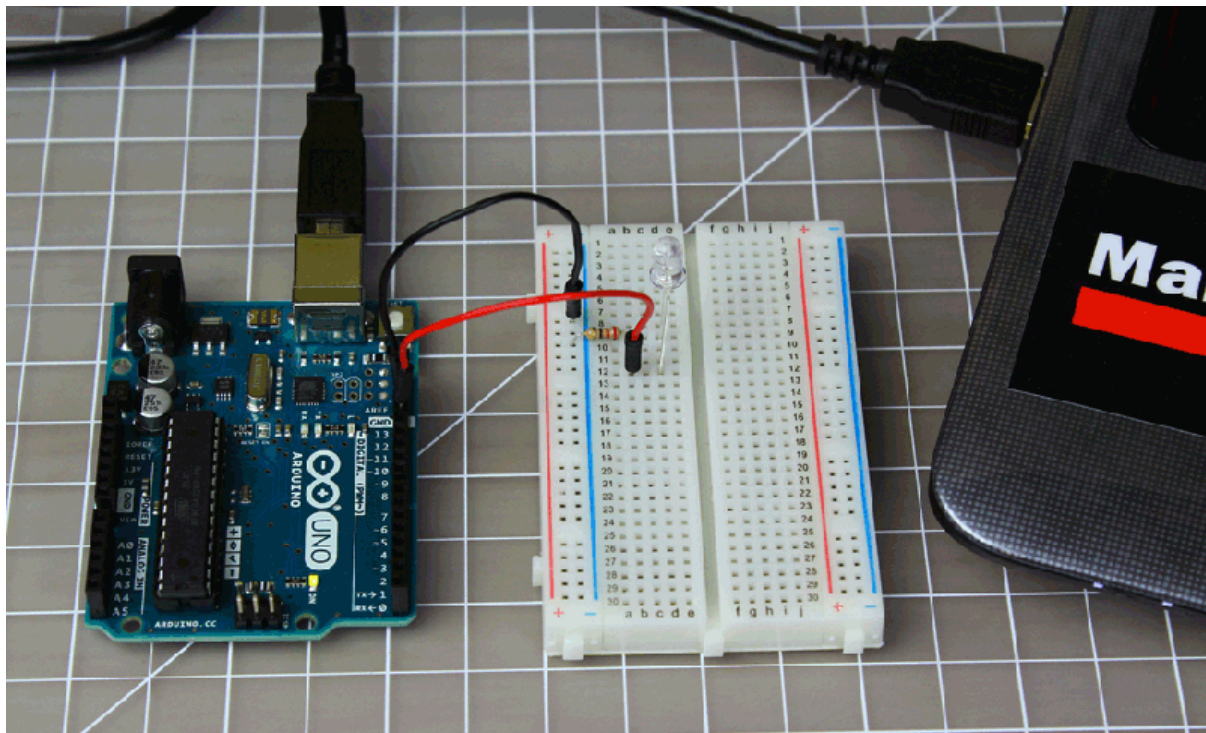
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```


Next, you need to click on the verify button (check mark) that's located in the top left of the IDE box. This will compile the sketch and look for errors. Once it says "Done Compiling" you are ready to upload it. Click the upload button (forward arrow) to send the program to the Arduino board.



The built-in LEDs on the Arduino board will flash rapidly for a few seconds and then the program will execute. If everything went correctly, the LED on the breadboard should turn on for a second and then off for a second and continue in a loop.



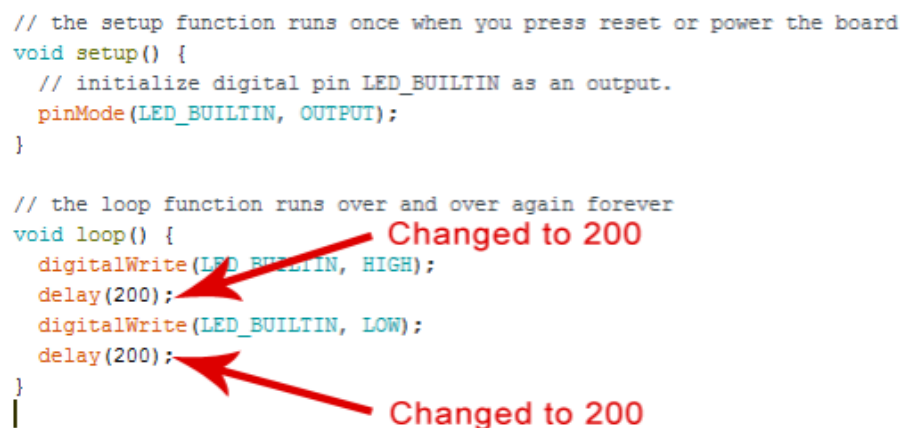
Congrats! You just completed your first Arduino project. **Troubleshooting** – If you ran into a problem don't give up, check out the troubleshooting section at the end of the chapter for common problems.

Change The Code

Before we go to the next project, let's change some of the code in the "Blink" sketch to make it do something different. Playing around with the sketch will help you start to learn how the code controls the board.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(200);
  digitalWrite(LED_BUILTIN, LOW);
  delay(200);
}
```

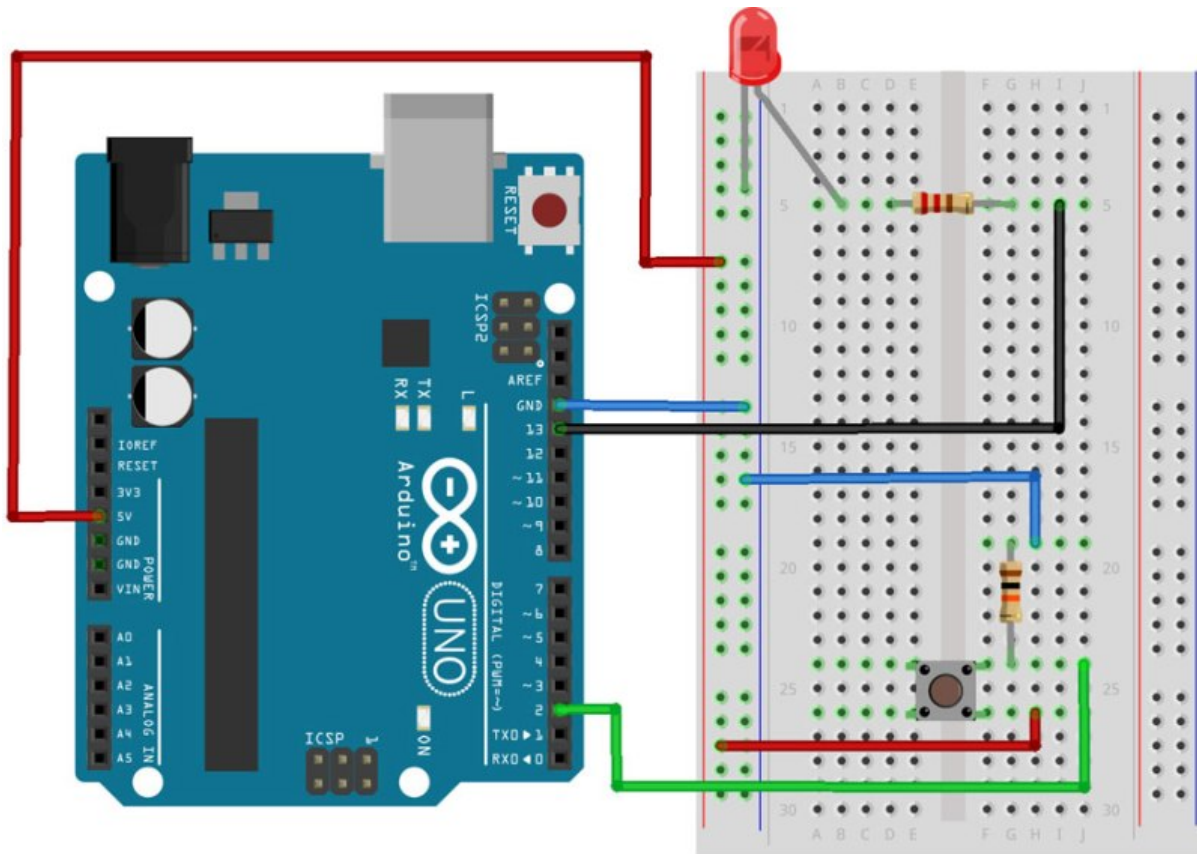
The diagram shows the Arduino code with two red arrows pointing to the delay values. The first arrow points to the first 'delay(200);' line, and the second arrow points to the second 'delay(200);' line. Both arrows are labeled with the text 'Changed to 200' in red.

Keep the Arduino board connected and change the delay portion of the code from (1000) to (200). Click the verify button on the top left of the IDE and then click upload. This should make the LED on the breadboard blink faster.

NOTE – Arduino measures time in milliseconds and 1000 milliseconds = 1 second. The original code (1000) turns on the LED for 1 second and then off for 1 second. By adjusting the code from (1000) to (200) it shortens the time between on and off which makes it blink faster.

Arduino Project 2: LED w/ Switch

Now it's time to add a switch to your Arduino project. A switch is an electrical component that completes a circuit when pushed and breaks the circuit when released. In this project, we will be using a small pushbutton switch to control an LED.



Required Parts

- [Arduino Uno Board](#)
- [Breadboard](#) – half size
- [Jumper Wires](#)
- [USB Cable](#)
- [LED \(5mm\)](#)
- [Push button switch](#)
- [10k Ohm Resistor](#)
- [220 Ohm Resistor](#)

Connect The Parts

You can build your Arduino circuit by looking at the breadboard image or by using the written description below. In the written description, we will use a letter/number combo that refers to the location of the component. If we mention H19 for example, that refers to column H, row 19 on the breadboard.

Step 1 – Connect the blue jumper wire from the GND on the Arduino to the GND rail (blue line) on the breadboard near A13

Step 2 – Connect the blue jumper wire from the GND rail on the breadboard near A17 to H19

Step 3 – Connect the red jumper wire from the power rail on the breadboard around row A27 to H26

Step 4 – Connect the green jumper wire from pin 2 on Arduino to J24 on the breadboard

Step 5 – Place one leg of a 10k Ohm resistor in G19 and the other leg in G24

Step 6 – Place the pushbutton switch into F24, F26, E24 and E26

Step 7 – Place one leg of a 220 Ohm resistor in D5 and the other leg in G5

Step 8 – Insert the short leg of the LED in the GND rail around A5 and the long leg in B5

Step 9 – Connect the black jumper wire from pin 13 on the Arduino to I5 on the breadboard

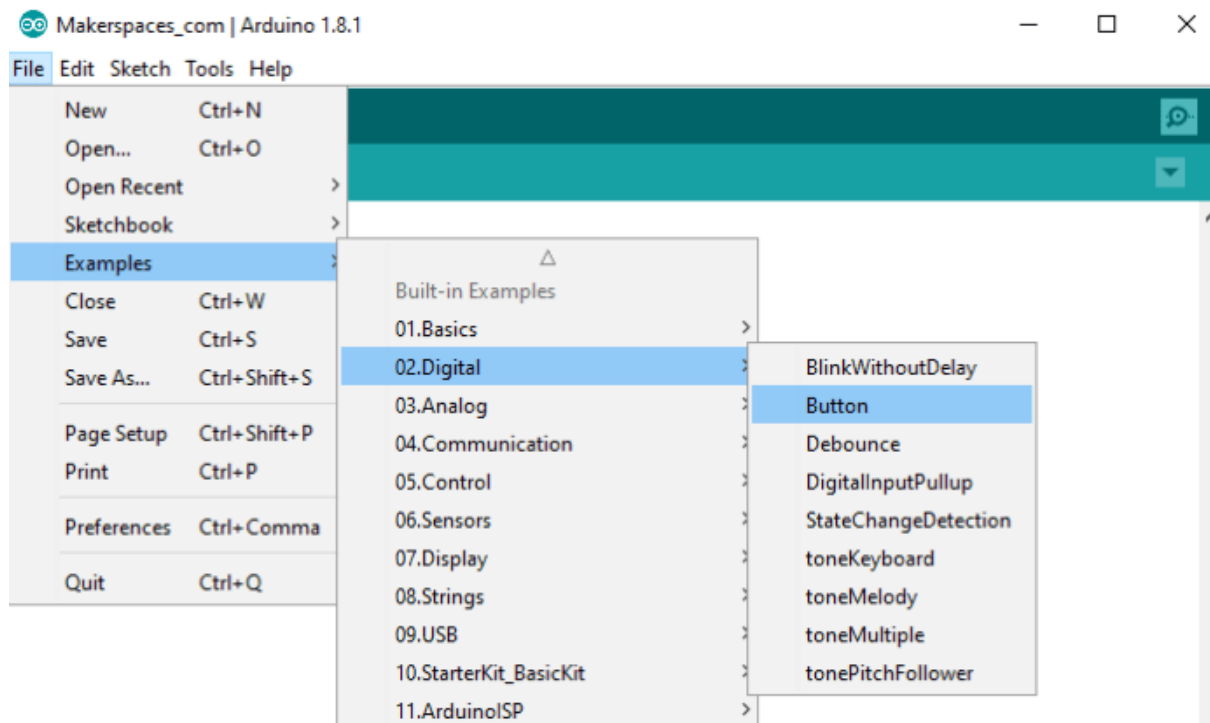
Step 10 – Connect the red jumper wire from 5V on the Arduino to power rail (+) near A8

Step 11 – Connect the Arduino Uno to your computer via USB cable

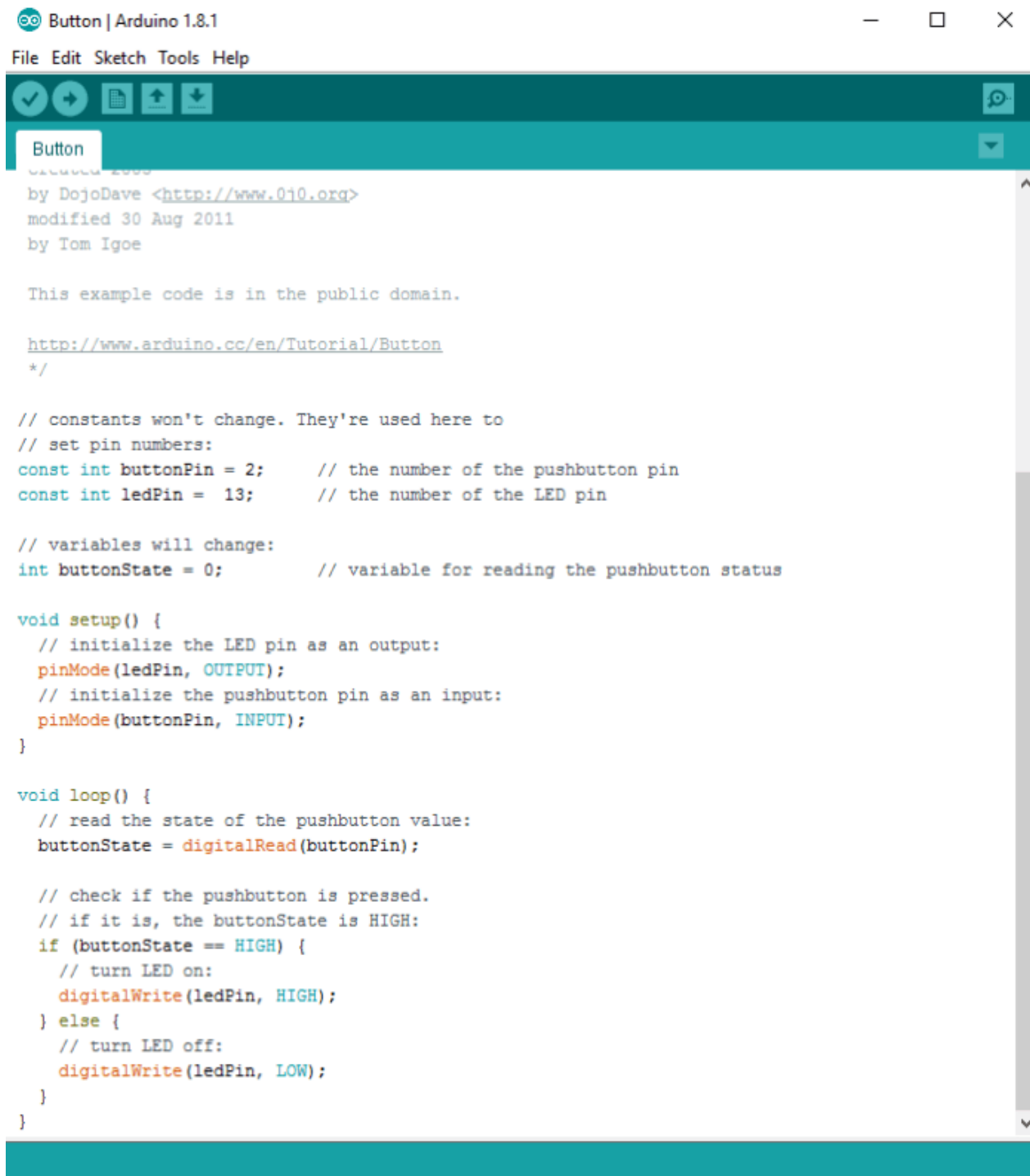
Upload The Switch Sketch

Now it's time to upload the sketch to the Arduino that will allow us to use a switch. As with the blink sketch, there are example programs already loaded in the Arduino IDE that we'll be using.

In order to use a switch, we have to load the file called "Button" which can be found at this location in the IDE: **File > Examples > Digital > Button**



Now you should have a fully coded button sketch that looks like the image below.

A screenshot of the Arduino IDE interface. The title bar reads "Button | Arduino 1.8.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, undo, redo, and other functions. A dropdown menu is open, showing the name of the sketch, "Button". The main text area contains the following code:

```
/*  
  Button  
  by DojoDave <http://www.0j0.org>  
  modified 30 Aug 2011  
  by Tom Igoe  
  
  This example code is in the public domain.  
  
  http://www.arduino.cc/en/Tutorial/Button  
  */  
  
// constants won't change. They're used here to  
// set pin numbers:  
const int buttonPin = 2;    // the number of the pushbutton pin  
const int ledPin = 13;      // the number of the LED pin  
  
// variables will change:  
int buttonState = 0;        // variable for reading the pushbutton status  
  
void setup() {  
  // initialize the LED pin as an output:  
  pinMode(ledPin, OUTPUT);  
  // initialize the pushbutton pin as an input:  
  pinMode(buttonPin, INPUT);  
}  
  
void loop() {  
  // read the state of the pushbutton value:  
  buttonState = digitalRead(buttonPin);  
  
  // check if the pushbutton is pressed.  
  // if it is, the buttonState is HIGH:  
  if (buttonState == HIGH) {  
    // turn LED on:  
    digitalWrite(ledPin, HIGH);  
  } else {  
    // turn LED off:  
    digitalWrite(ledPin, LOW);  
  }  
}
```

Click on the verify button (check mark) that's located in the top left of the IDE box. Once it says "Done Compiling" you are ready to upload it. Click the upload button (forward arrow) to send the program to the Arduino board.

Press the button switch on the breadboard and you should be able to turn on and off the LED.

Troubleshooting

If you are having any problems with the projects we did, make sure the following has been checked.

1. Verify the LED is actually functional. Use a 3v coin cell battery and connect the LONG leg of the LED to the (+) and SHORT leg to the (-) of the battery.
2. Verify the correct leg of the LED is connected properly. LONG leg to positive and SHORT leg to negative.
3. Make sure the Arduino IDE shows the correct board. Go to **Tools > Board** then select **Arduino Uno**.
4. Make sure the Arduino IDE shows the correct port. Go to **Tools > Port** then select the port that says **Arduino**.
5. Verify all component connections are secure with the Arduino board and breadboard.

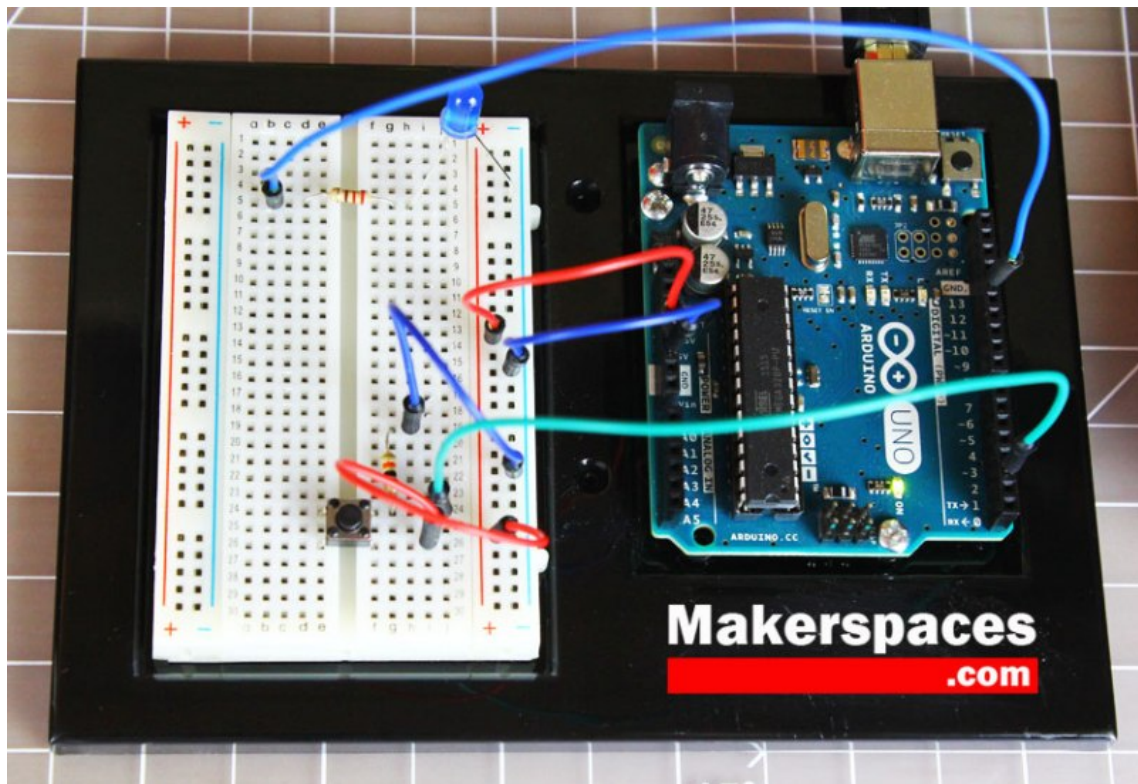
6

**Arduino
Practice
Projects**

Arduino Practice Projects

Now it's time to expand your knowledge by creating circuits using the following electronic components:

- LED
- RGB LED
- Temp Sensor
- Pushbutton
- Potentiometer
- Photoresistor
- Servo
- Motor
- Buzzer
- LCD screen



Download The Arduino Project Code (APC)

In order to complete the 15 projects in this chapter, you will need to download the Arduino Project Code (APC). Use the link below to download the zip folder containing the code.

Download Project Code – www.Makerspaces.com/APC

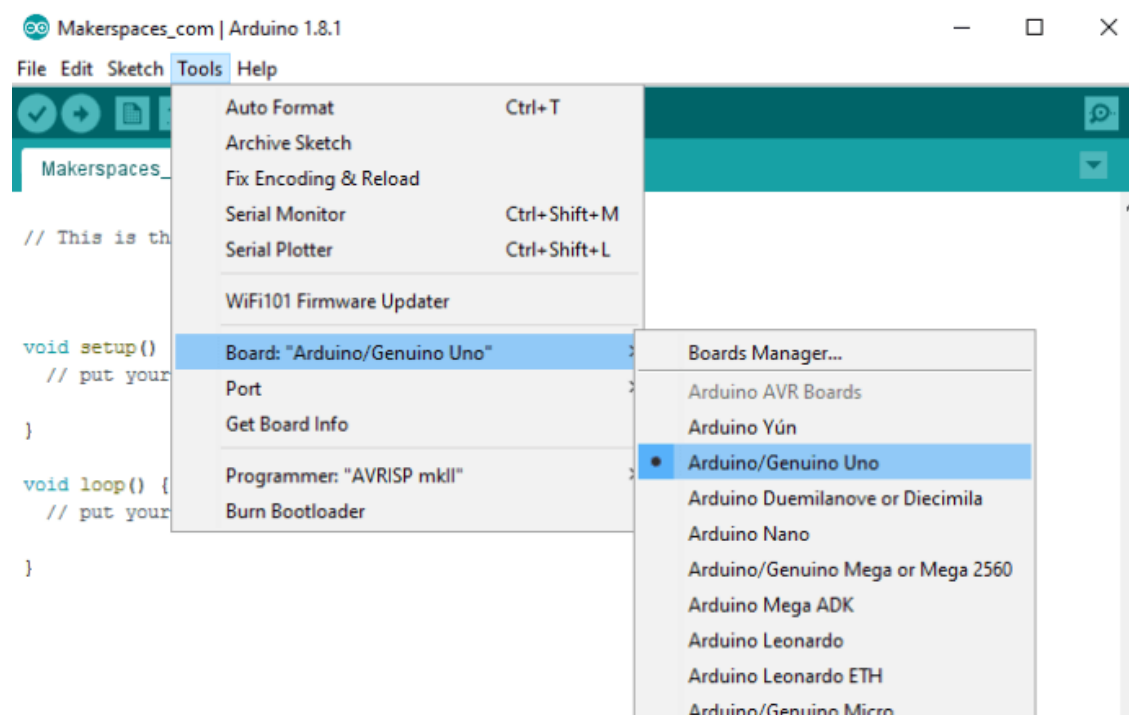
Once the file has been downloaded, you will need to unzip/extract the folder in order to use it.

Connect & Configure The Arduino Uno

Before you can start working with Arduino, you will need to make sure you have the IDE software installed on your computer. You can download the IDE for free on Arduino's website as outlined in an earlier chapter.

Select The Board

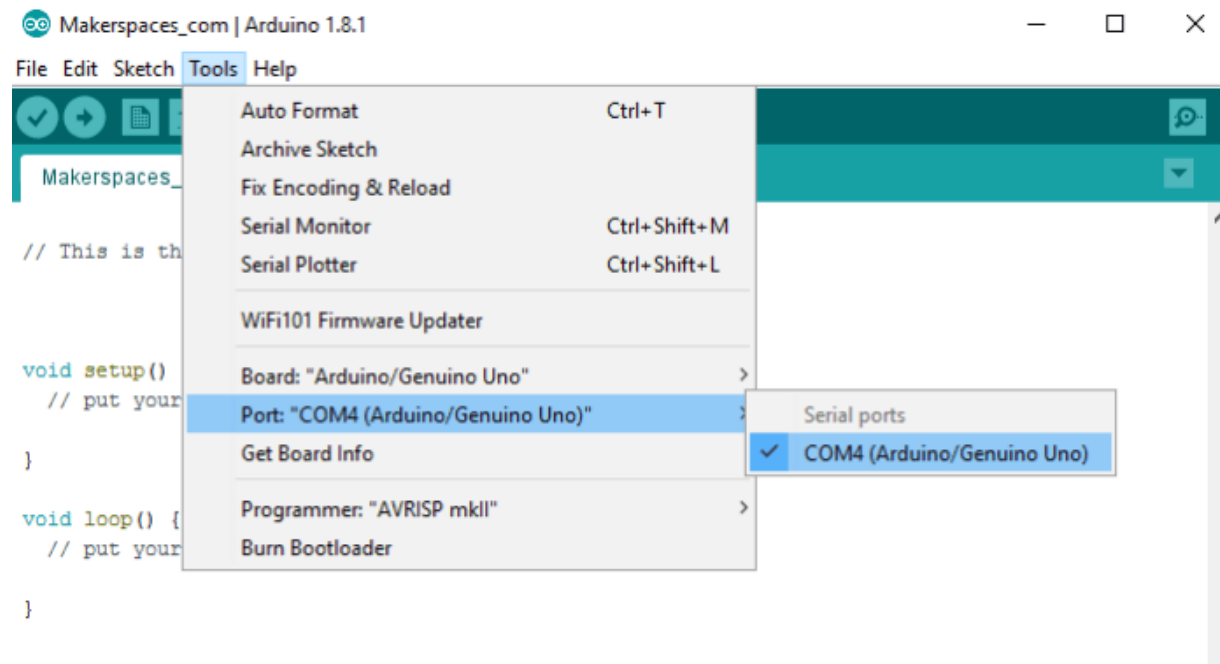
Connect the Arduino to the computer using the USB cable. Once plugged in, you will need to open the IDE and click on **Tools > Board > Arduino Uno** to select the board.



Select Serial Port

Next, you have to select the port that you are using on your computer.

To select the port, go to **Tools > Port** and then select the COM that says **Arduino**.



At this point, you are ready to create the project circuits as outlined in this chapter.

Project 1 – Test The Arduino

The first project is one of the most basic and simple circuits you can create with Arduino. This project will test your Arduino by blinking an LED that is connected directly to the board.

Parts Needed

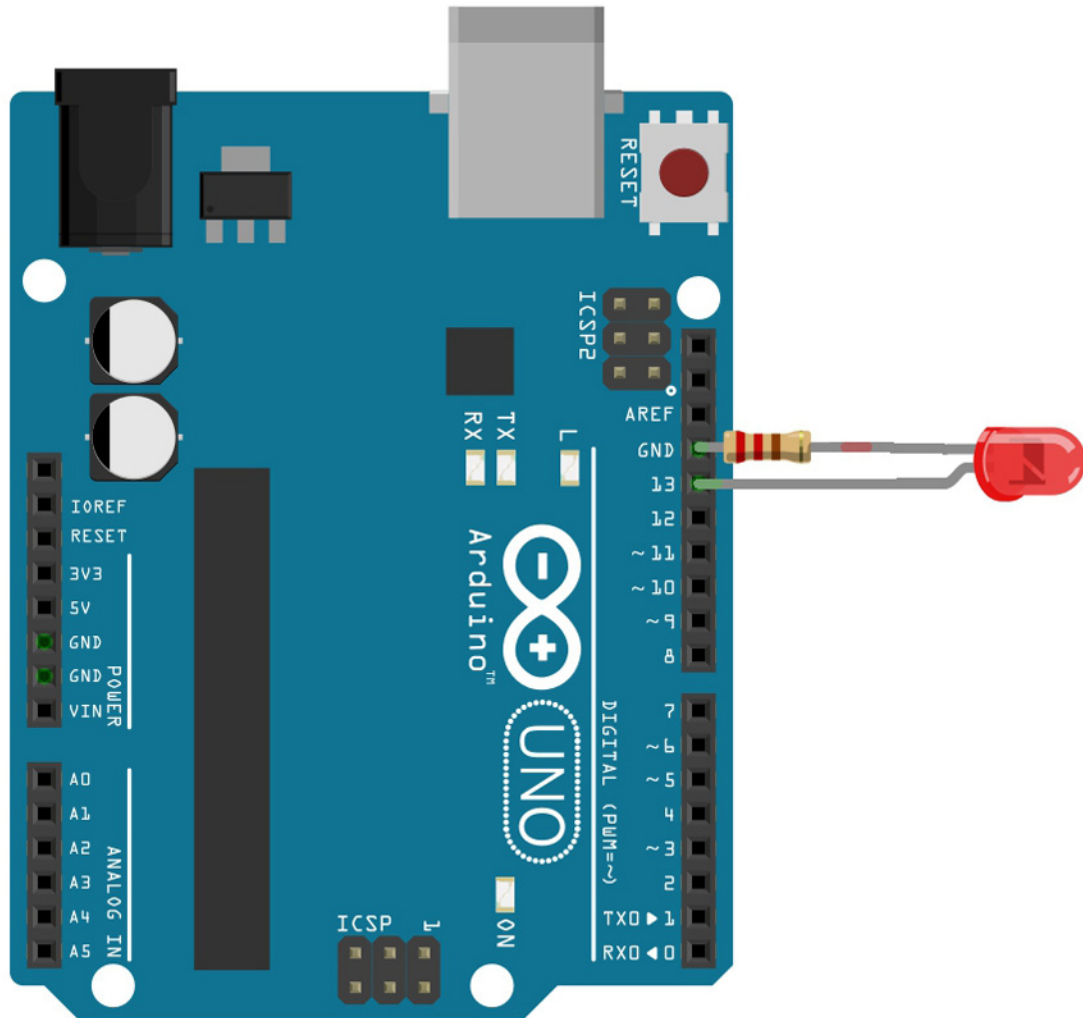
- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) LED 5mm
- (1) 220 Ω Resistor

Project Steps

1. Twist a 220 Ω resistor to the long leg (+) of the LED.
2. Push the short leg of the LED into the ground (GND) pin on the board.
3. Push the resistor leg that's connected to the LED into the #13 pin.

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_01_TestArduino**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
LED (5mm)	twist (-) leg to resistor		(+) leg Pin 13
Resistor 220 Ohm			GND

Component	Breadboard		Arduino UNO

Project 2 – Blink An LED

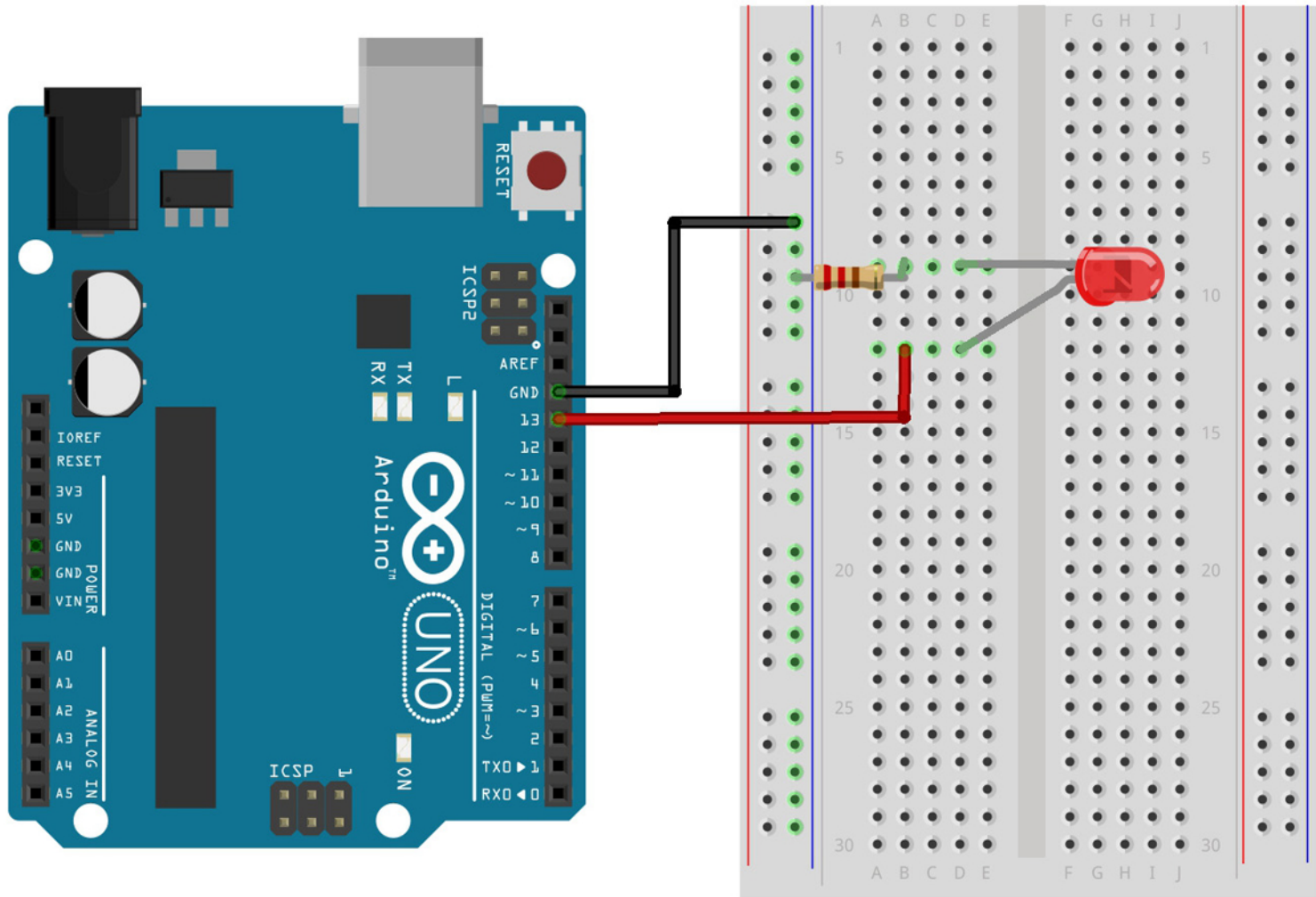
This project is identical to project #1 except that we will be building it on a breadboard. Once complete, the LED should turn on for a second and then off for a second in a loop.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (2) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_02_Blink**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
LED (5mm)	- Leg D 9	+ Leg D 12	
Resistor 220 Ohm	B 9	(-) Rail	
Jumper (RED)	B 12		Pin 13
Jumper (BLACK)	(-) Rail		GND

Component	Breadboard		Arduino UNO

Project 3 – Pushbutton

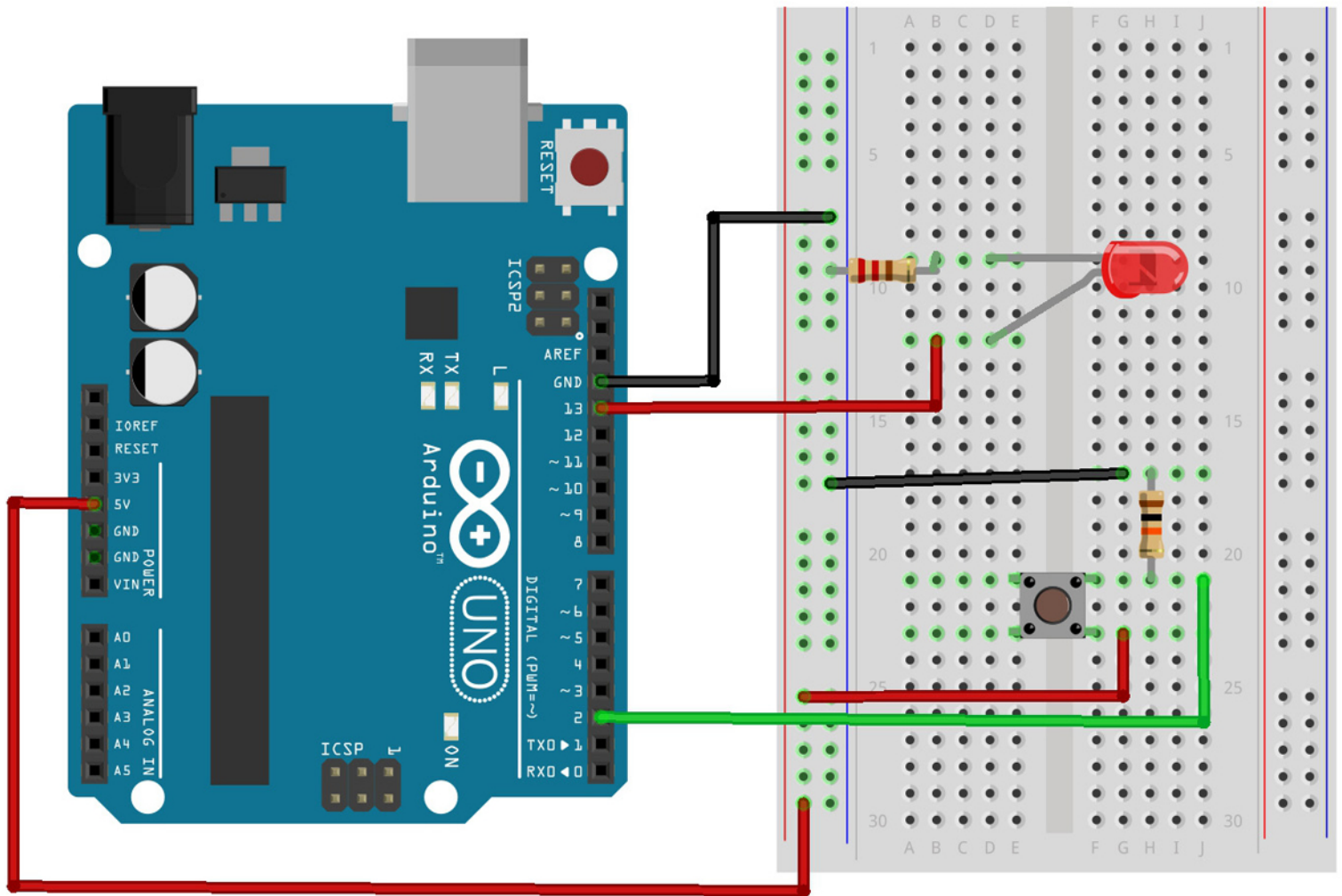
Using a push button switch, you will be able to turn on and off an LED.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (1) 10K Ω Resistor
- (1) Push Button Switch
- (6) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_03_Pushbutton**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
LED (5mm)	- Leg D 9	+ Leg D 12	
Resistor 220 Ohm	B 9	(-) Rail	
Resistor 10K Ohm	H 17	H 21	
Push Button Switch	F21, F23	E21, E23	
Jumper (RED)	(+) Rail		5 V
Jumper (BLACK)	(-) Rail		GND
Jumper (RED)	B 12		Pin 13
Jumper (GREEN)	J 21		Pin 2
Jumper (RED)	(+) Rail	G 23	
Jumper (BLACK)	(-) Rail	G 17	

Component	Breadboard		Arduino UNO

Project 4 – Potentiometer

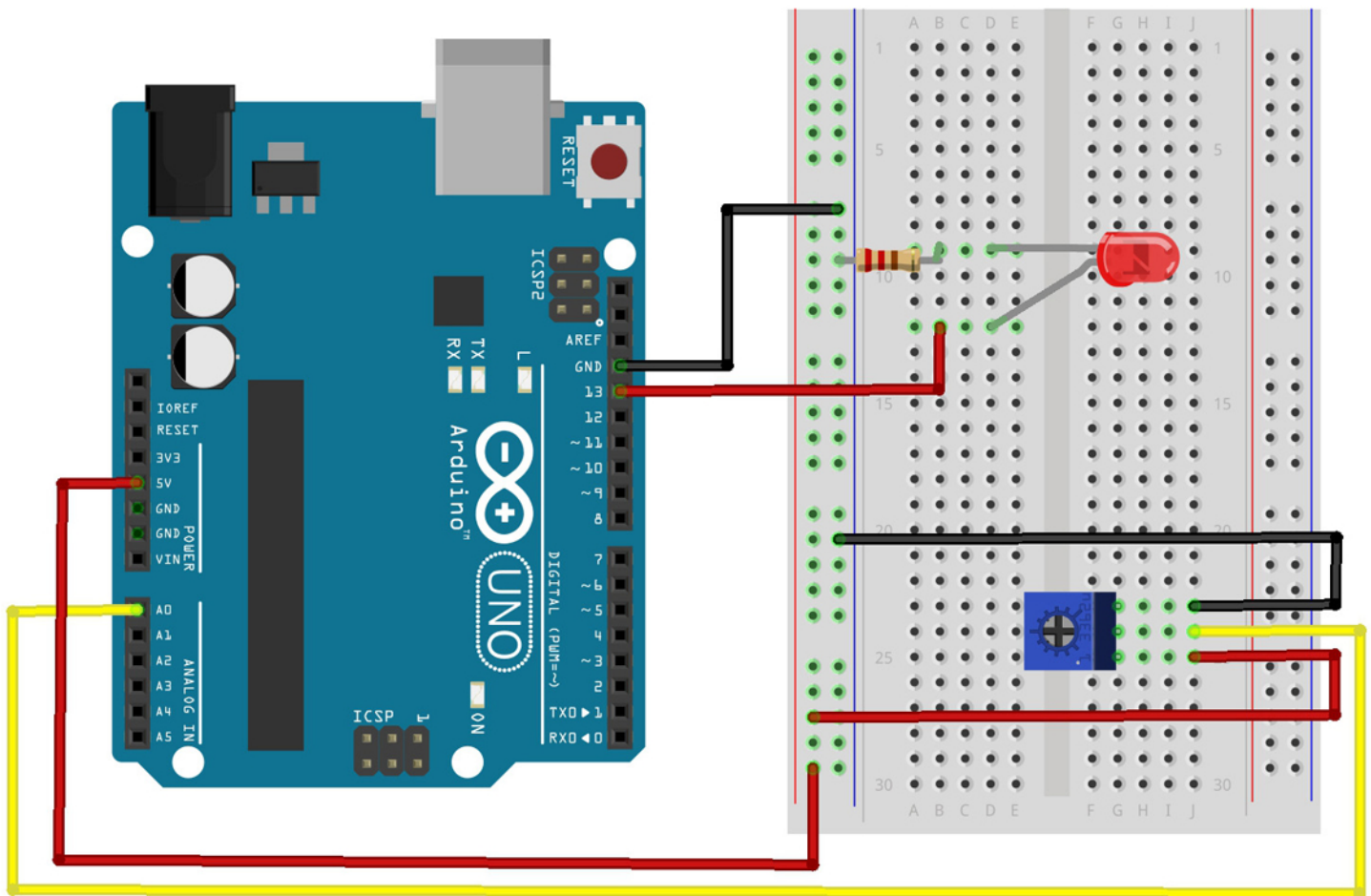
Using a potentiometer, you will be able to control the resistance of an LED. Turning the knob will increase and decrease the frequency the LED blinks.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (1) Potentiometer (10k Trimpot)
- (6) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_04_Potentiometer**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
LED (5mm)	- Leg D 9	+ Leg D 12	
Resistor 220 Ohm	B 9	(-) Rail	
Potentiometer	G23, G24, G25		
Jumper (BLACK)	(-) Rail		GND
Jumper (RED)	B 12		Pin 13
Jumper (RED)	(+) Rail		5 V
Jumper (YELLOW)	J 24		A 0
Jumper (RED)	(+) Rail	J 25	
Jumper (BLACK)	(-) Rail	J 23	

Component	Breadboard		Arduino UNO

Project 5 – Fade An LED

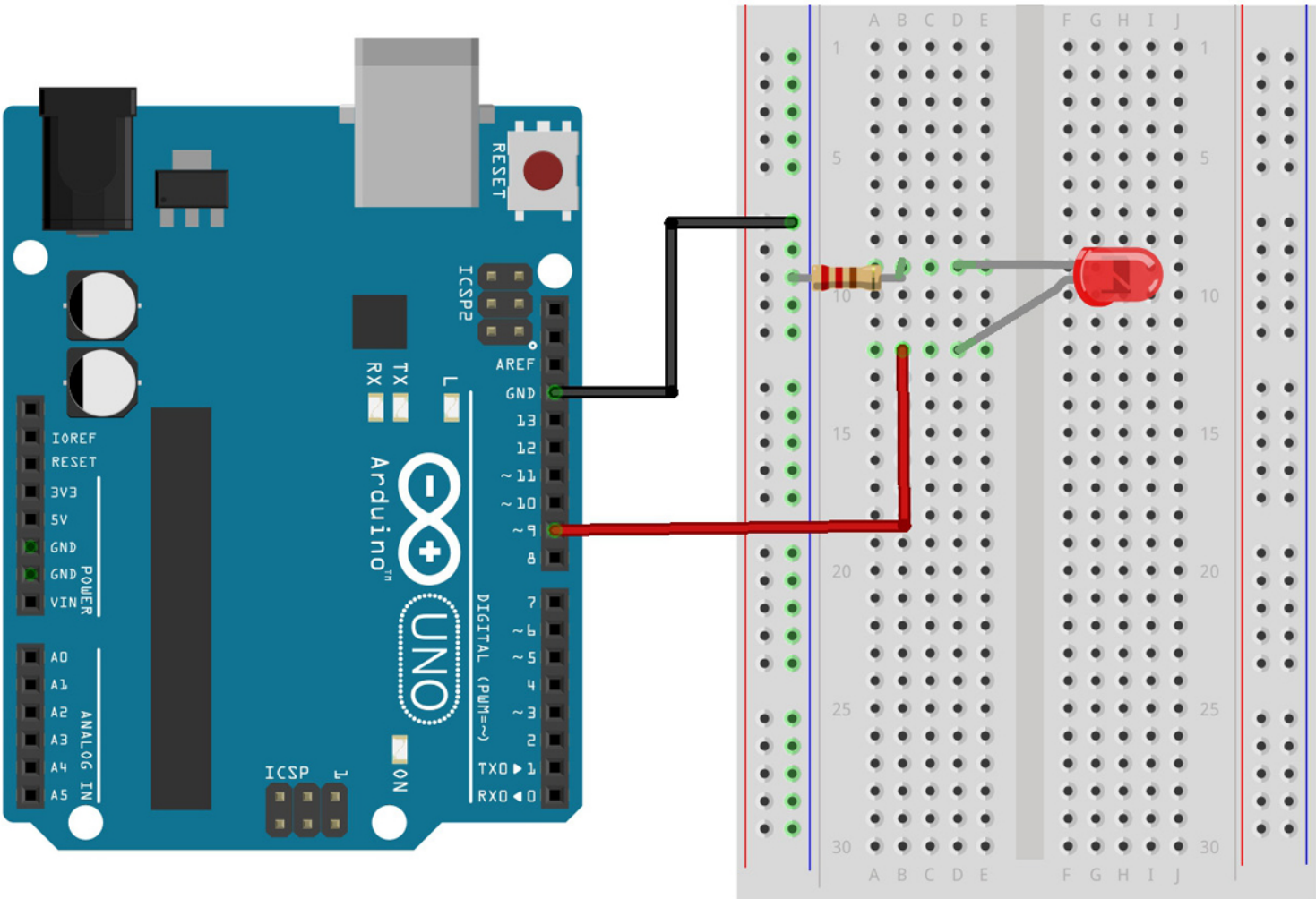
By using a PWM pin on the Arduino, you will be able to increase and decrease the intensity of brightness of an LED.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (2) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_05_Fade**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
LED (5mm)	- Leg D 9	+ Leg D 12	
Resistor 220 Ohm	B 9	(-) Rail	
Jumper (RED)	B 12		Pin 9
Jumper (BLACK)	(-) Rail		GND

Component	Breadboard		Arduino UNO

Project 6 – Scrolling LED

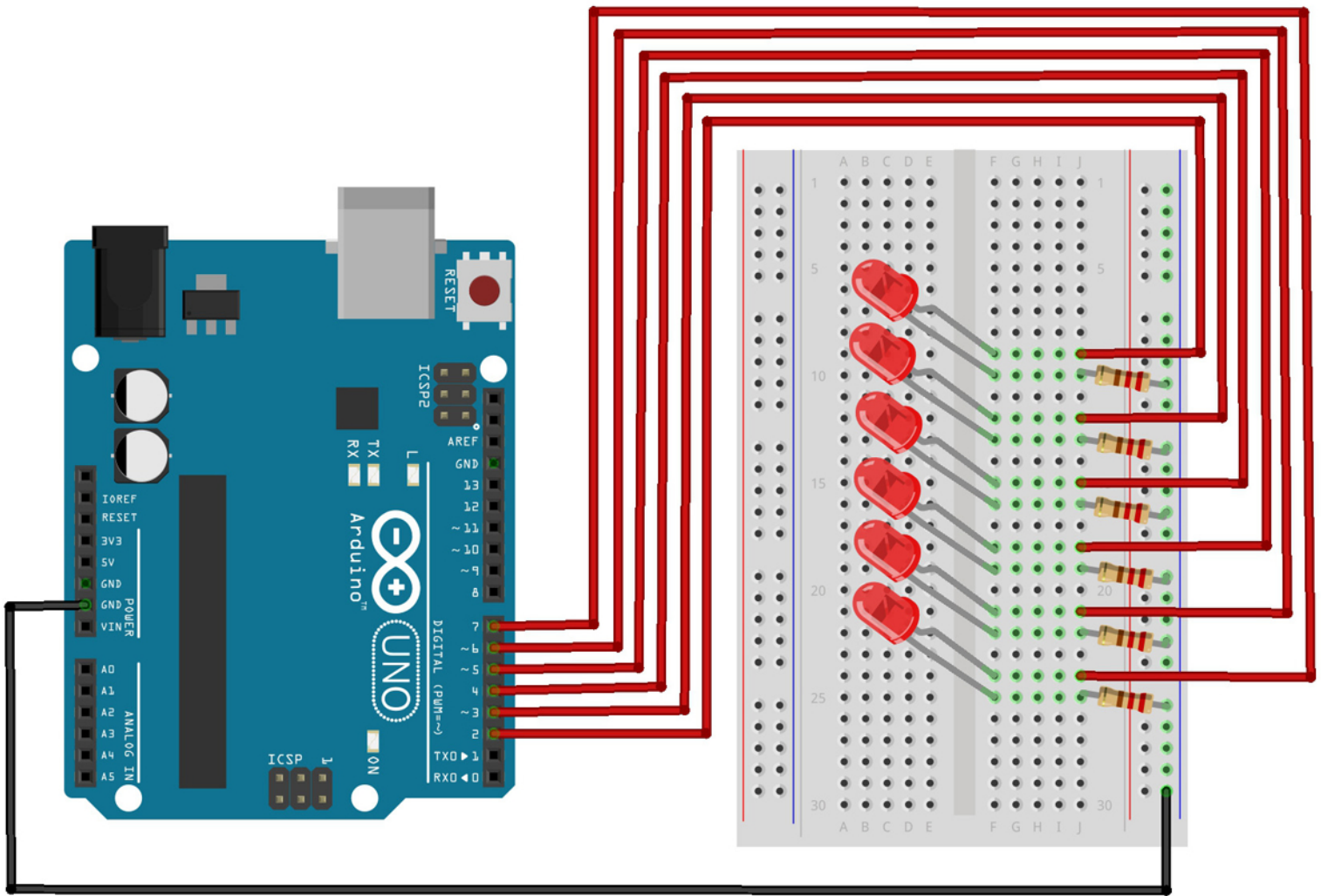
This project will blink 6 LEDs, one at a time, in a back and forth formation. This type of circuit was made famous by the show Knight Rider which featured a car with looping LEDs.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (6) LED 5mm
- (6) 220 Ω Resistor
- (7) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_06_Scrolling**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
LED (5mm)	- Leg F 10	+ Leg F 9	
LED (5mm)	- Leg F 13	+ Leg F 12	
LED (5mm)	- Leg F 16	+ Leg F 15	
LED (5mm)	- Leg F 19	+ Leg F 18	
LED (5mm)	- Leg F 22	+ Leg F 21	
LED (5mm)	- Leg F 25	+ Leg F 24	
Resistor 220 Ohm	J 10	(-) Rail	
Resistor 220 Ohm	J 13	(-) Rail	
Resistor 220 Ohm	J 16	(-) Rail	
Resistor 220 Ohm	J 19	(-) Rail	
Resistor 220 Ohm	J 22	(-) Rail	
Resistor 220 Ohm	J 25	(-) Rail	

Component	Breadboard		Arduino UNO
Jumper (BLACK)	(-) Rail		GND
Jumper (RED)	J 24		Pin 7
Jumper (RED)	J 21		Pin 6
Jumper (RED)	J 18		Pin 5
Jumper (RED)	J 15		Pin 4
Jumper (RED)	J 12		Pin 3
Jumper (RED)	J 9		Pin 2

Project 7 – Bargraph

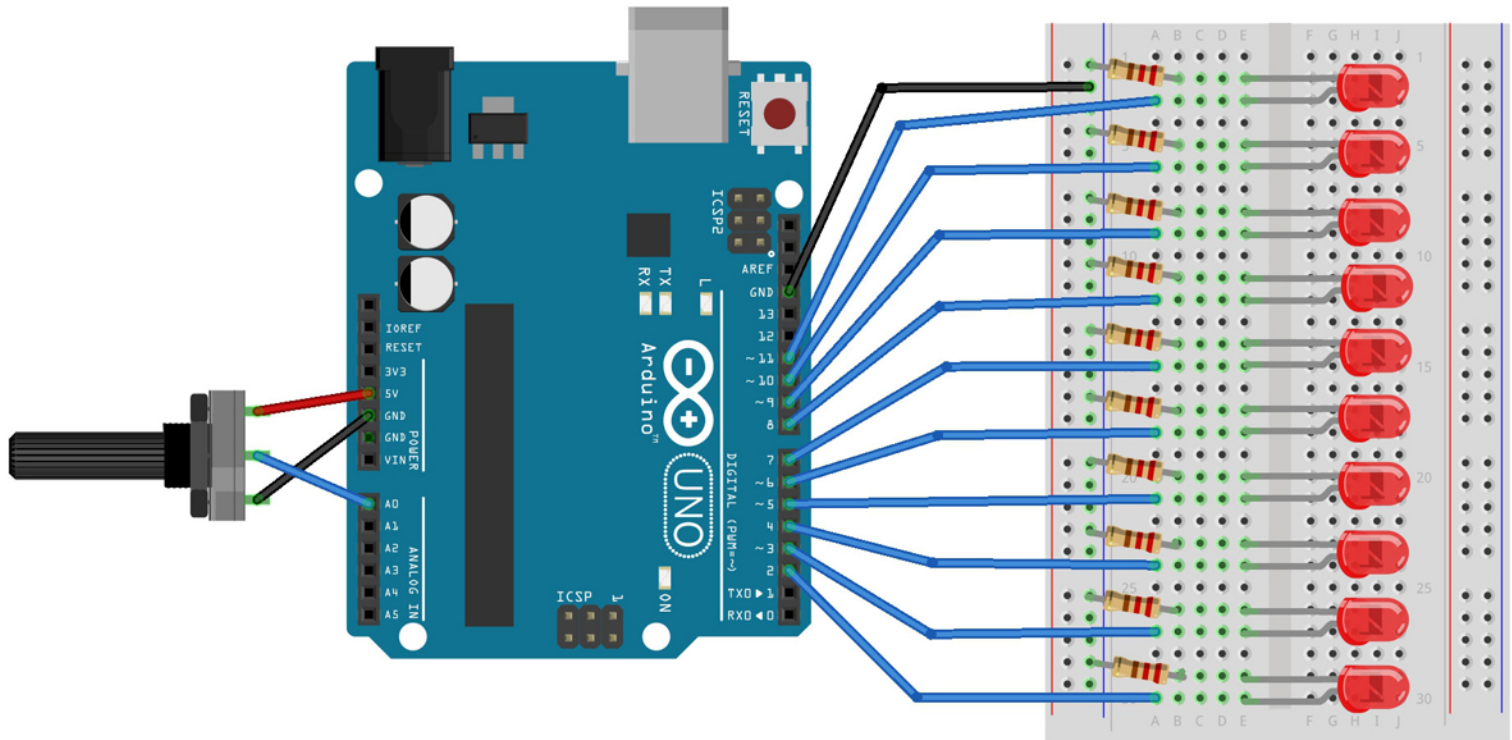
Using a potentiometer, you can control a series of LEDs in a row. Turning the potentiometer knob will turn on or off more of the LEDs.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) Potentiometer – Rotary
- (10) LED 5mm
- (10) 220 Ω Resistor
- (11) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_07_BarGraph**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
Potentiometer			5V, A0, GND
LED (5mm)	(-) Leg E 2	(+) Leg E 3	
LED (5mm)	(-) Leg E 5	(+) Leg E 6	
LED (5mm)	(-) Leg E 8	(+) Leg E 9	
LED (5mm)	(-) Leg E 11	(+) Leg E 12	
LED (5mm)	(-) Leg E 14	(+) Leg E 15	
LED (5mm)	(-) Leg E 17	(+) Leg E 18	
LED (5mm)	(-) Leg E 20	(+) Leg E 21	
LED (5mm)	(-) Leg E 23	(+) Leg E 24	
LED (5mm)	(-) Leg E 26	(+) Leg E 27	
LED (5mm)	(-) Leg E 29	(+) Leg E 30	
Resistor 220 Ohm	(-) Rail	B 2	
Resistor 220 Ohm	(-) Rail	B 5	
Resistor 220 Ohm	(-) Rail	B 8	
Resistor 220 Ohm	(-) Rail	B 11	
Resistor 220 Ohm	(-) Rail	B 14	

Component	Breadboard		Arduino UNO
Resistor 220 Ohm	(-) Rail	B 17	
Resistor 220 Ohm	(-) Rail	B 20	
Resistor 220 Ohm	(-) Rail	B 23	
Resistor 220 Ohm	(-) Rail	B 26	
Resistor 220 Ohm	(-) Rail	B 29	
Jumper (BLACK)	(-) Rail		GND
Jumper (BLUE)	A 3		Pin 11
Jumper (BLUE)	A 6		Pin 10
Jumper (BLUE)	A 9		Pin 9
Jumper (BLUE)	A 12		Pin 8
Jumper (BLUE)	A 15		Pin 7
Jumper (BLUE)	A 18		Pin 6
Jumper (BLUE)	A 21		Pin 5
Jumper (BLUE)	A 24		Pin 4
Jumper (BLUE)	A 27		Pin 3
Jumper (BLUE)	A 30		Pin 2

Project 8 – Multiple LEDs

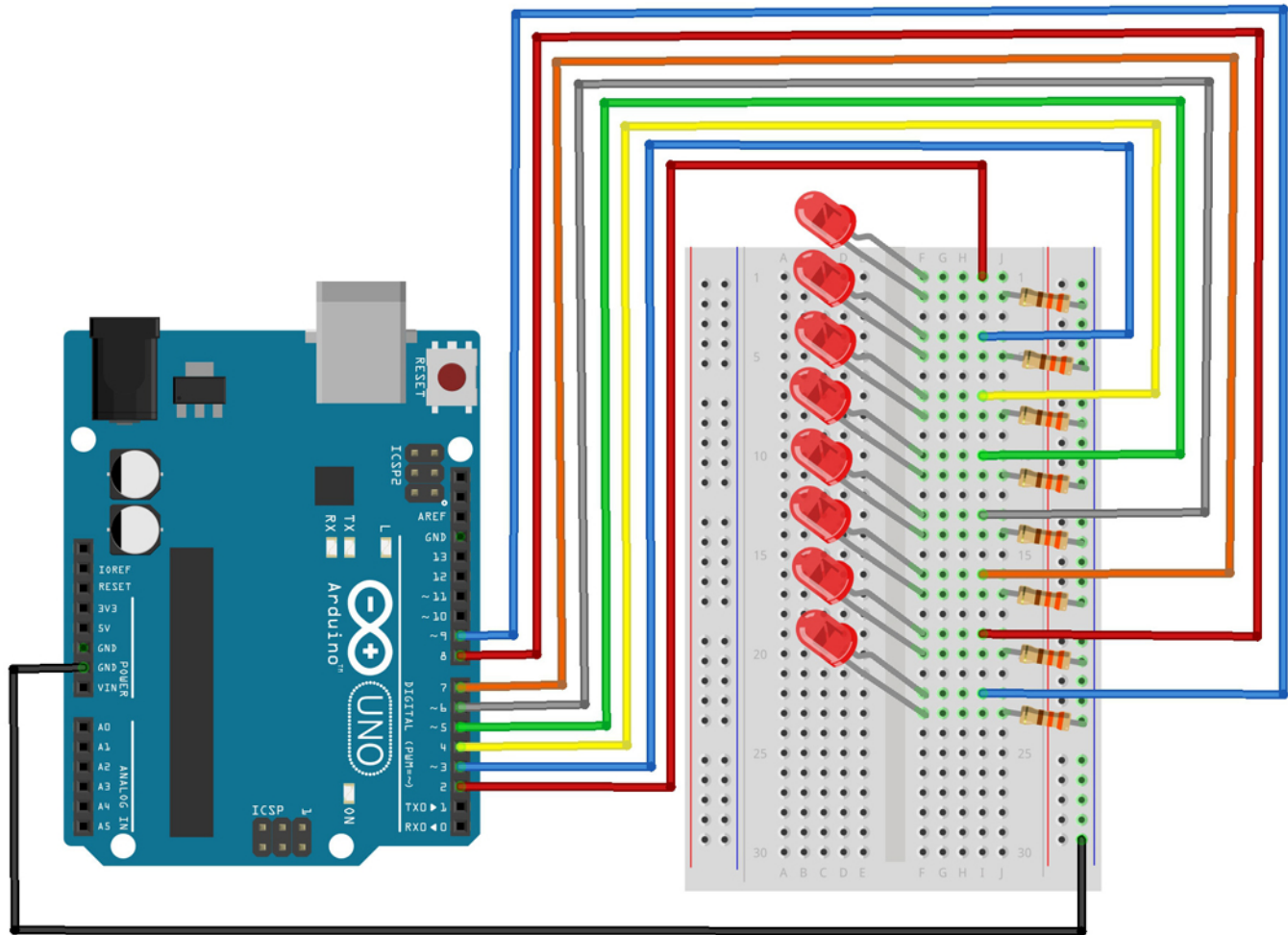
This project will use 8 pins on the Arduino board to blink 8 LEDs at the same time.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (8) LED 5mm
- (8) 330 Ω Resistor
- (9) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_08_MultipleLEDs**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
LED (5mm)	(-) Leg F 2	(+) Leg F 1	
LED (5mm)	(-) Leg F 5	(+) Leg F 4	
LED (5mm)	(-) Leg F 8	(+) Leg F 7	
LED (5mm)	(-) Leg F 11	(+) Leg F 10	
LED (5mm)	(-) Leg F 14	(+) Leg F 13	
LED (5mm)	(-) Leg F 17	(+) Leg F 16	
LED (5mm)	(-) Leg F 20	(+) Leg F 19	
LED (5mm)	(-) Leg F 23	(+) Leg F 22	
Resistor 330 Ohm	(-) Rail	J 2	
Resistor 330 Ohm	(-) Rail	J 5	
Resistor 330 Ohm	(-) Rail	J 8	
Resistor 330 Ohm	(-) Rail	J 11	
Resistor 330 Ohm	(-) Rail	J 14	
Resistor 330 Ohm	(-) Rail	J 17	
Resistor 330 Ohm	(-) Rail	J 20	
Resistor 330 Ohm	(-) Rail	J 23	

Component	Breadboard		Arduino UNO
Jumper (BLACK)	(-) Rail		GND
Jumper (BLUE)	I 22		Pin 9
Jumper (RED)	I 19		Pin 8
Jumper (ORANGE)	I 16		Pin 7
Jumper (GREY)	I 13		Pin 6
Jumper (GREEN)	I 10		Pin 5
Jumper (YELLOW)	I 7		Pin 4
Jumper (BLUE)	I 4		Pin 3
Jumper (RED)	I 1		Pin 2

Project 9 – RGB LED

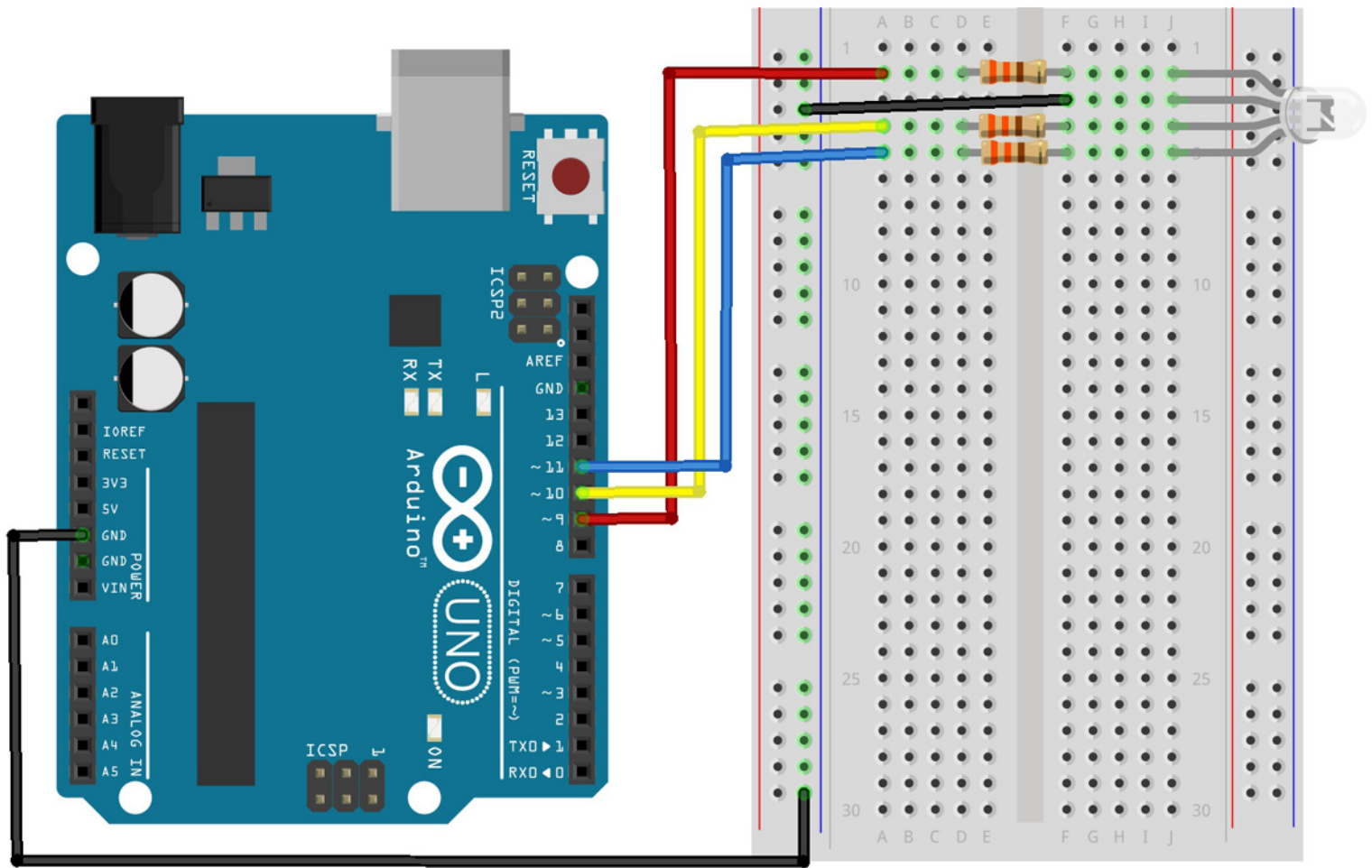
This project will be using an RGB LED to scroll through a variety of colors. RGB stands for Red, Green and Blue and this LED has the ability to create nearly unlimited color combinations.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) RGB LED
- (3) 330 Ω Resistor
- (5) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_09_RGBLED**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
RGB LED	J2, J3, J4, J5		
Resistor 330 Ohm	F 2	D 2	
Resistor 330 Ohm	F 4	D 4	
Resistor 330 Ohm	F 5	D 5	
Jumper (RED)	A 2		Pin 9
Jumper (BLACK)	(-) Rail	F 3	
Jumper (BLACK)	(-) Rail		GND
Jumper (YELLOW)	A 4		Pin 10
Jumper (BLUE)	A 5		Pin 11

Component	Breadboard		Arduino UNO

Project 10 – Photoresistor

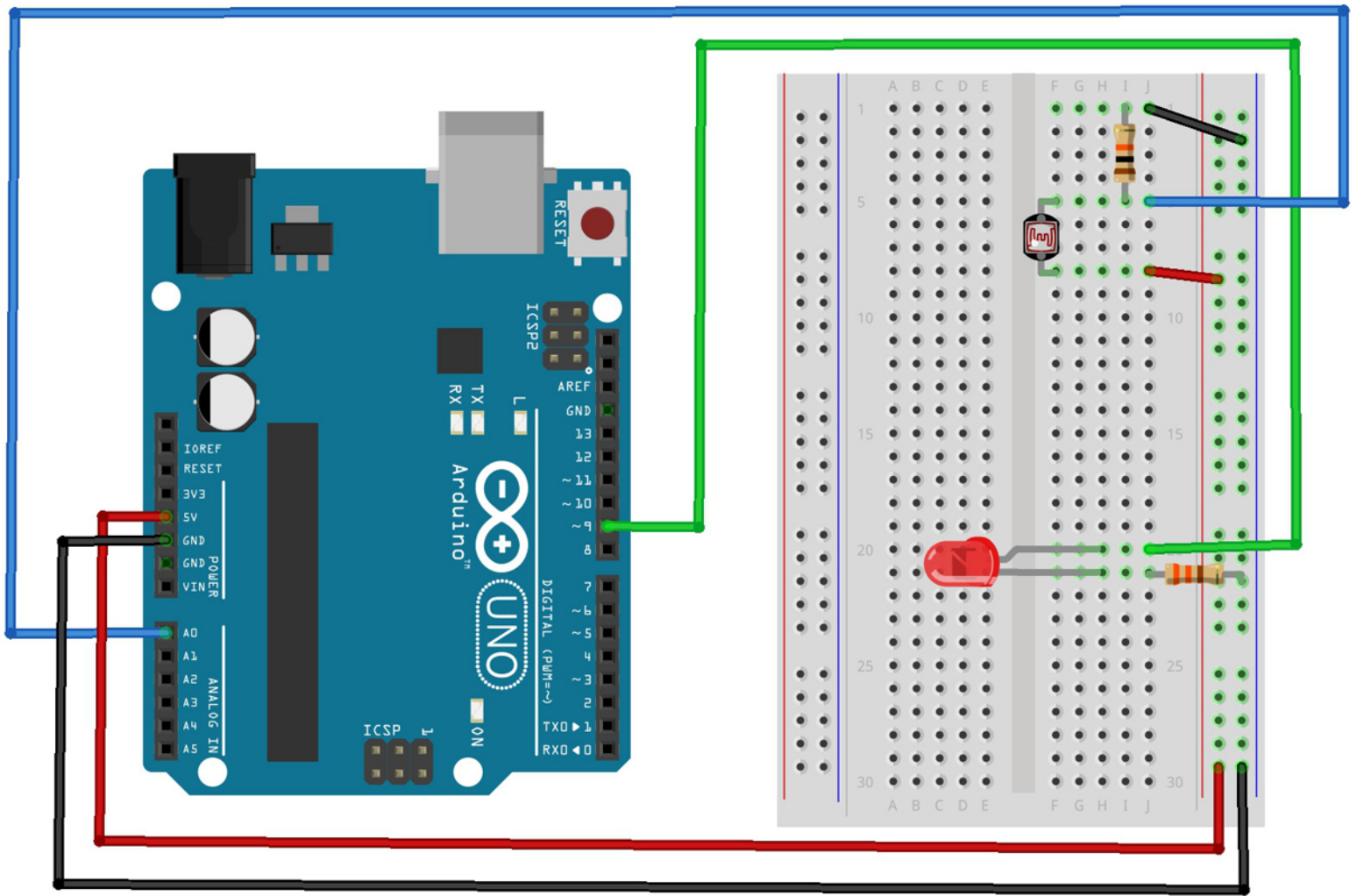
A photoresistor changes the resistance a circuit gets based on the amount of light that hits the sensor. In this project, the brightness of the LED will increase and decrease based on the amount of light present.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 330 Ω Resistor
- (1) 10K Ω Resistor
- (1) Photoresistor
- (6) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_10_Photoresistor**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
LED (5mm)	(-) Leg H21	(+) Leg H20	
Resistor 330 Ohm	(-) Rail	J 21	
Resistor 10K Ohm	I 1	I 5	
Photoresistor	F 5	F 8	
Jumper (RED)	(+) Rail		5 V
Jumper (BLACK)	(-) Rail		GND
Jumper (RED)	(+) Rail	J 8	
Jumper (BLACK)	(-) Rail	J 1	
Jumper (BLUE)	J 5		A0
Jumper (GREEN)	J 20		Pin 9

Component	Breadboard		Arduino UNO

Project 11 – Temp Sensor

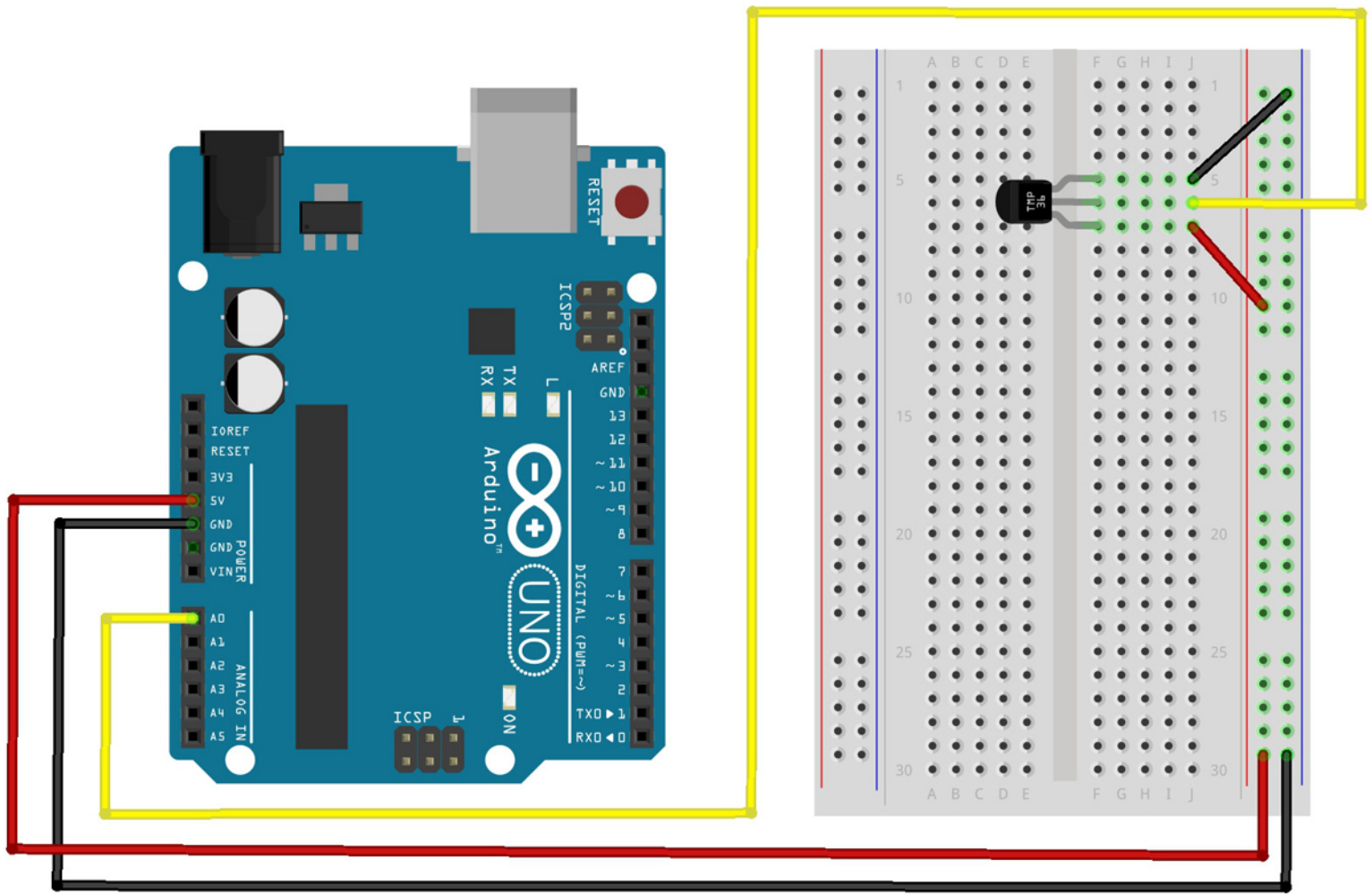
A temperature sensor measures ambient temperatures of the world around it. In this project, we will be displaying the temperature in the serial monitor of the Arduino IDE.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) Temperature Sensor – TMP36
- (5) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_11_TempSensor**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
Temp Sensor	F5, F6, F7		
Jumper (YELLOW)	J 6		A 0
Jumper (RED)	J 7	(+) Rail	
Jumper (BLACK)	J 5	(-) Rail	
Jumper (RED)	(+) Rail	5 V	
Jumper (BLACK)	(-) Rail	GND	

Component	Breadboard		Arduino UNO

Project 12 – Tone Melody

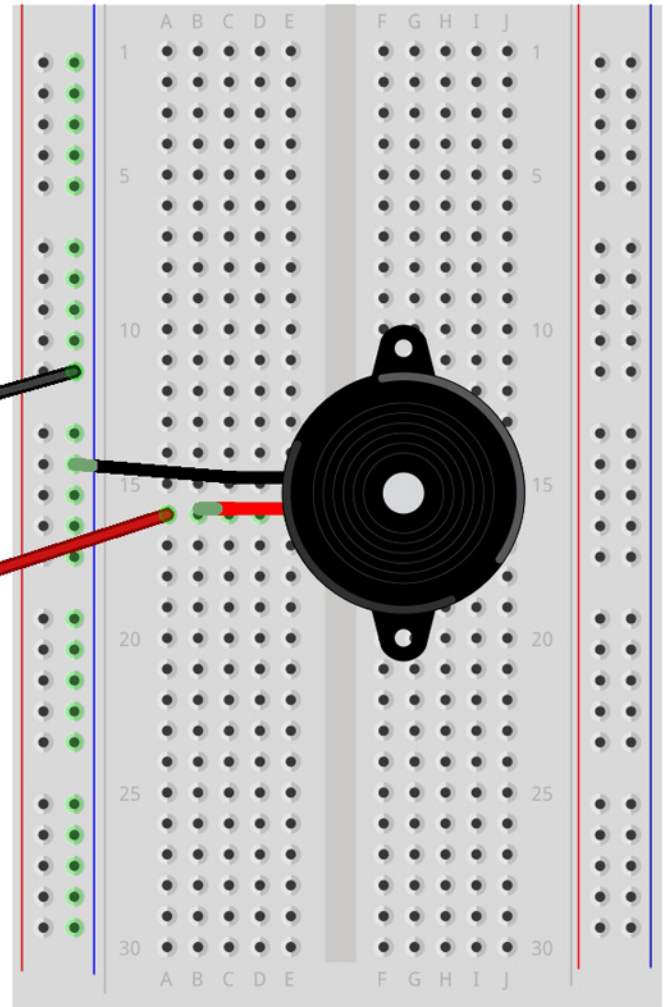
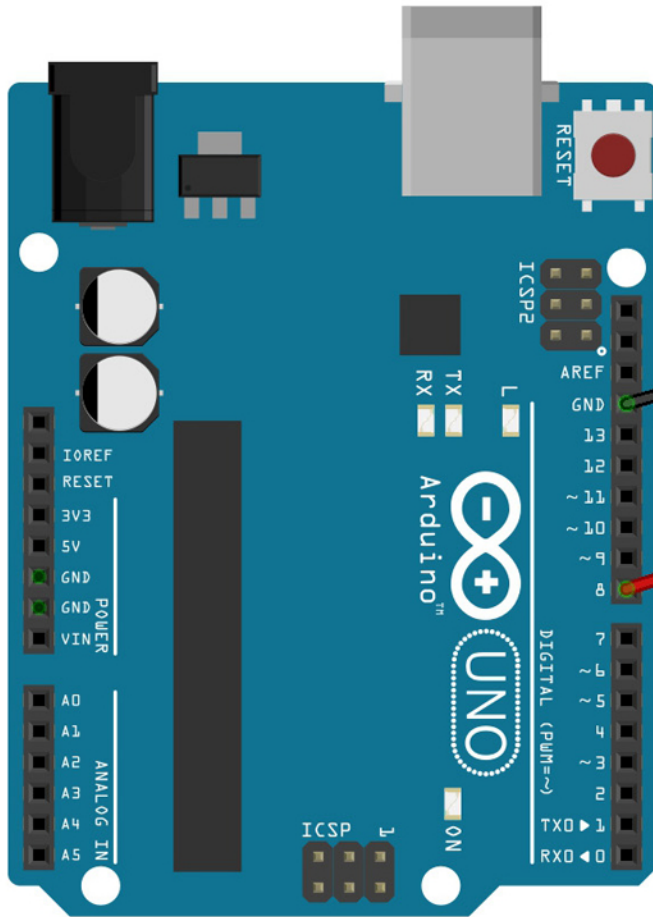
The project will use a piezo buzzer/speaker to play a little melody.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) Piezo Buzzer/Speaker
- (2) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_12_ToneMelody**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
Piezo Buzzer (red)	B 16		
Piezo Buzzer (black)	(-) Rail		
Jumper (BLACK)	(-) Rail		GND
Jumper (RED)	A 16		Pin 8

Component	Breadboard		Arduino UNO

Project 13 – Servo

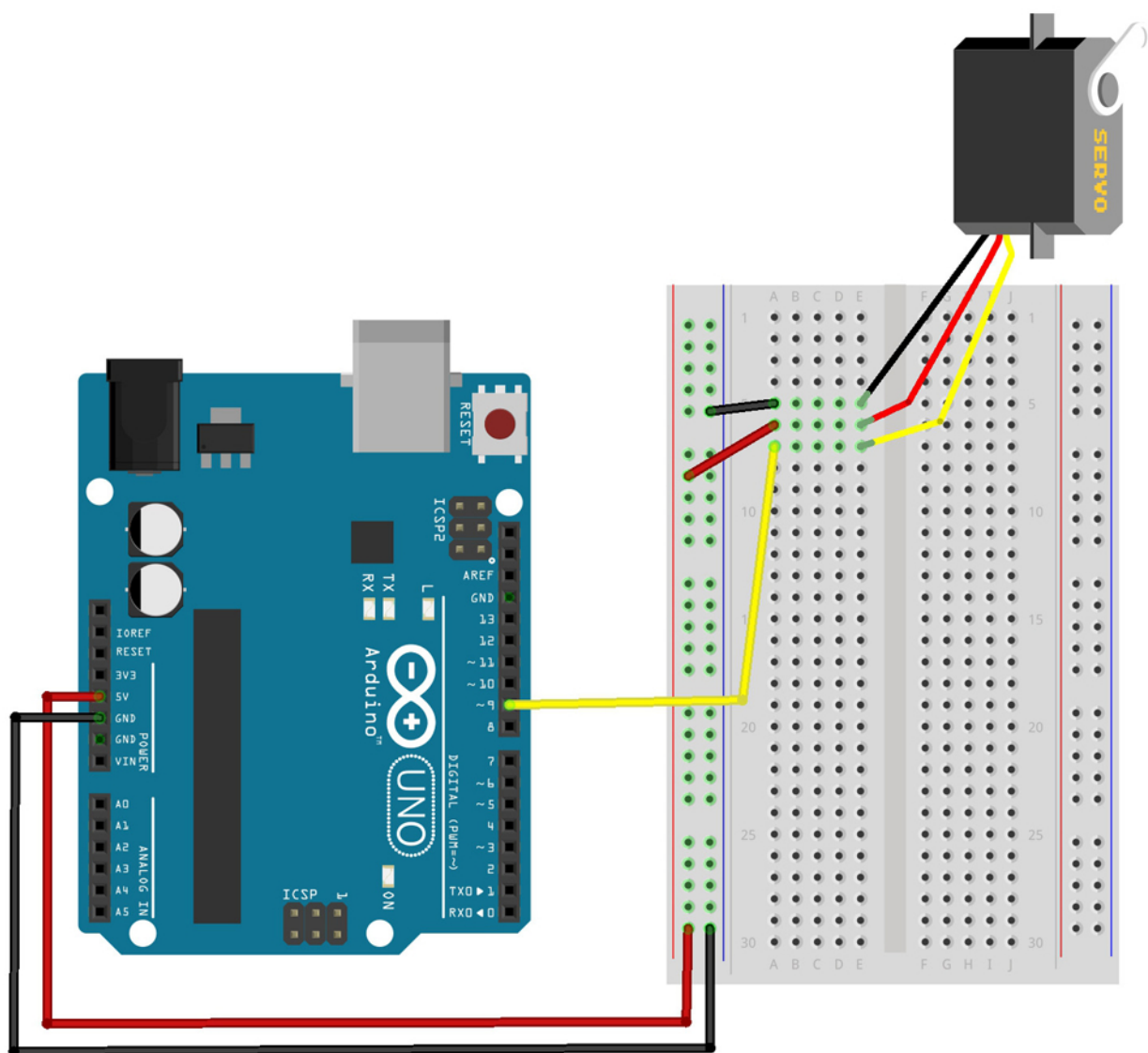
In this project, you will be able to sweep a servo back and forth through its full range of motion.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) Servo
- (6) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_13_Servo**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
Servo (Black)	E 5		
Servo (Red)	E 6		
Servo (Yellow)	E 7		
Jumper (RED)	A 6	(+) Rail	
Jumper (BLACK)	A 5	(-) Rail	
Jumper (YELLOW)	A 7		Pin 9
Jumper (RED)	(+) Rail		5 V
Jumper (BLACK)	(-) Rail		GND

Component	Breadboard		Arduino UNO

Project 14 – Motor

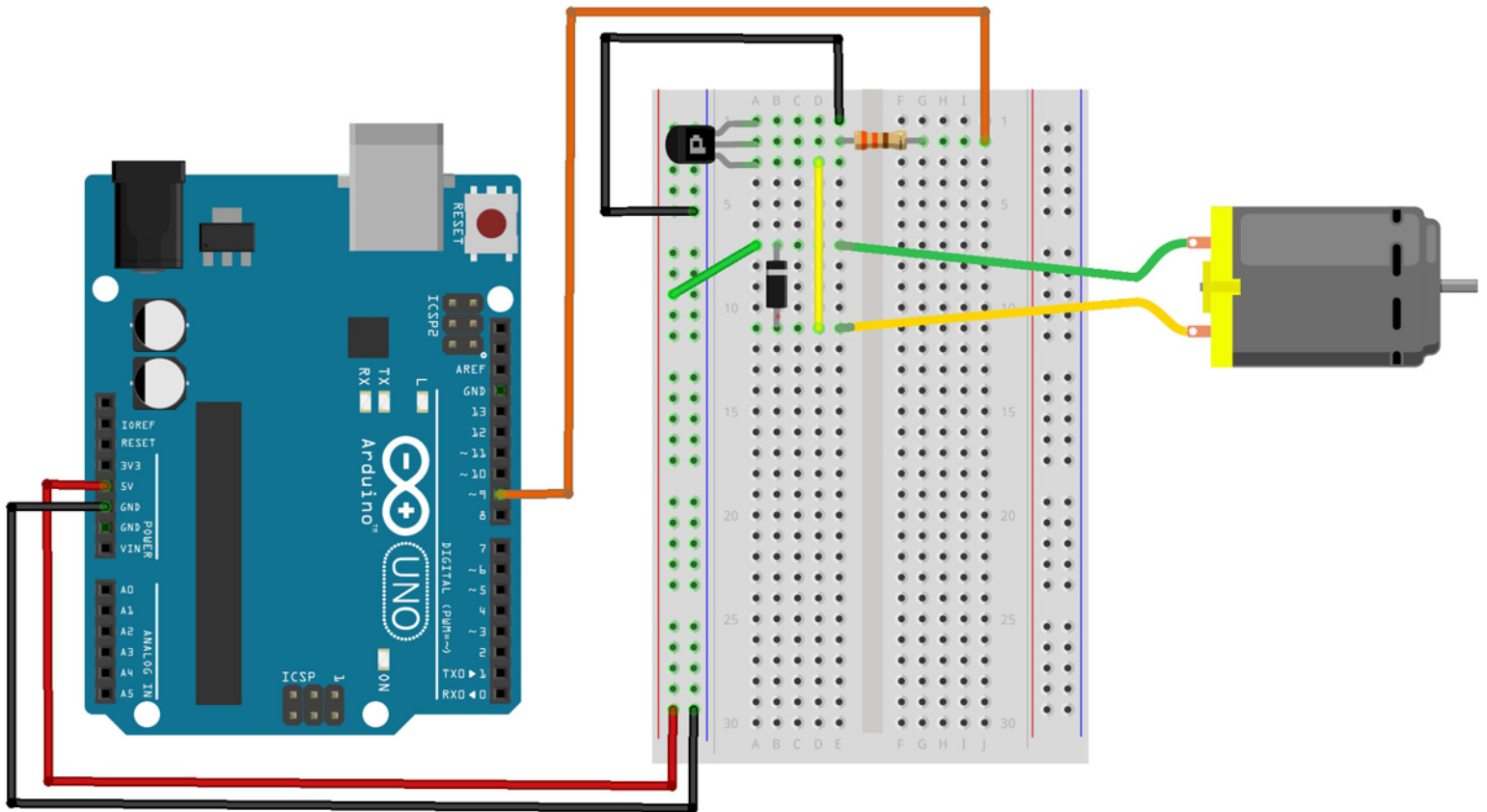
Using a switching transistor, we will be able to control a DC motor. If everything is connected correctly, you should see the motor spinning.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) DC Motor
- (1) 330 Ω Resistor
- (1) Diode 1N4148
- (1) NPN Transistor
- (6) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_14_Motor**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



Component	Breadboard		Arduino UNO
DC Motor - (Green)	E 7		
DC Motor - (Yellow)	E 11		
Diode - 1N4148	B 7	B 11	
Transistor - NPN	A1, A2, A3		
Resistor 330 Ohm	G 2	E 2	
Jumper (RED)	(+) Rail		5 V
Jumper (BLACK)	(-) Rail		GND
Jumper (ORANGE)	J 2		Pin 9
Jumper (GREEN)	A 7	(+) Rail	
Jumper (YELLOW)	D 3	D 11	
Jumper (BLACK)	(-) Rail	E 1	

Component	Breadboard		Arduino UNO

Project 15 – LCD Screen

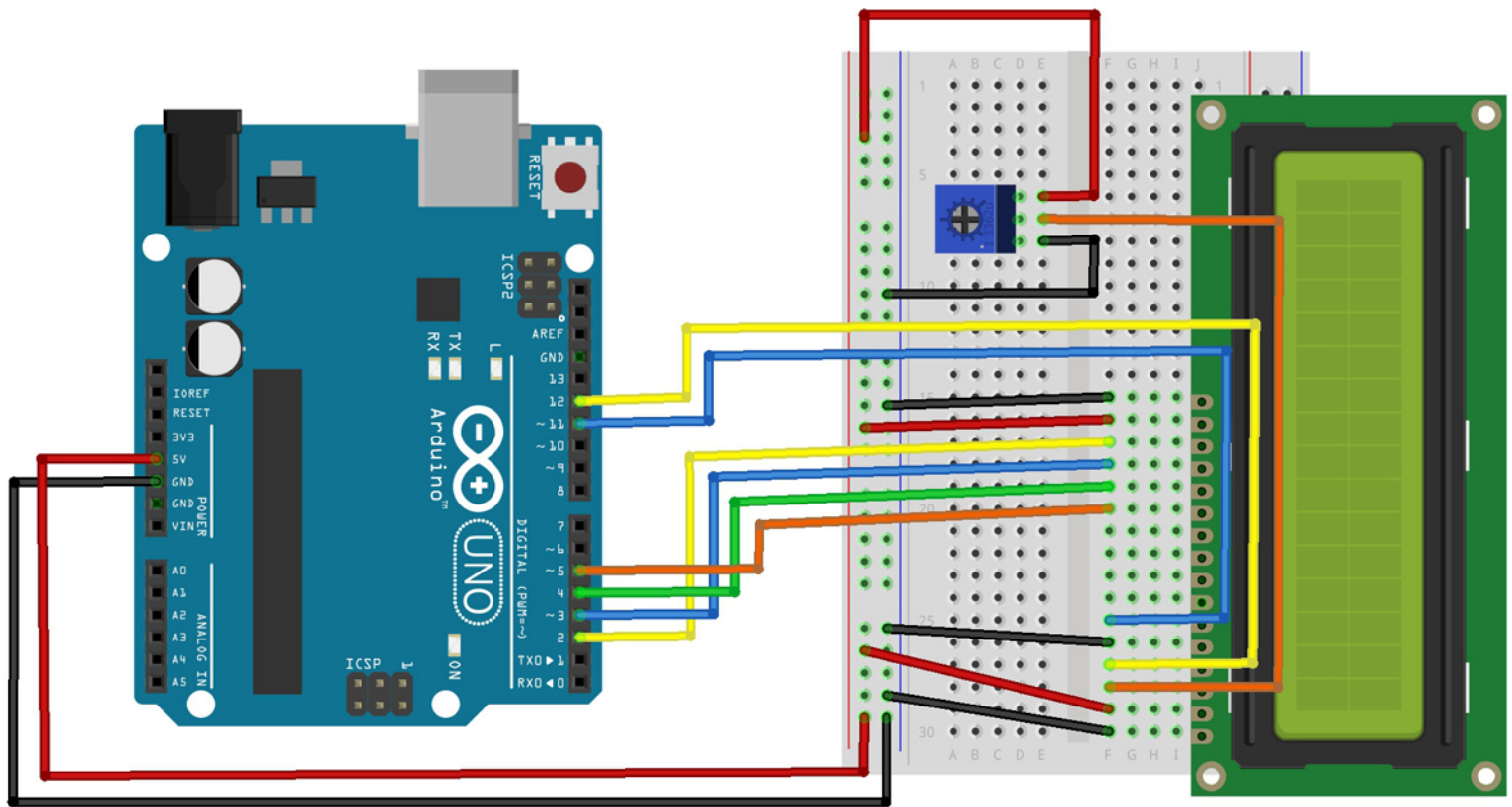
An LCD is a liquid crystal display that is able to display text on its screen. In this project, you should see the words “hello,world!” displayed on the screen. The potentiometer is used to adjust the contrast of the display.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LCD Screen
- (1) Potentiometer
- (16) Jumper Wires

Project Code

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_15_LCD**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.



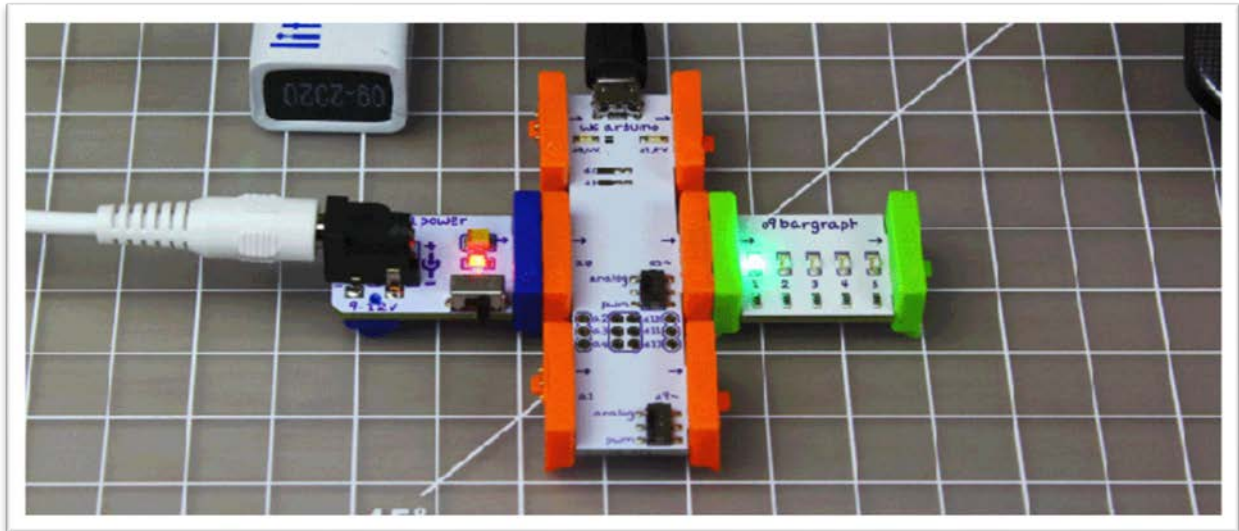
Component	Breadboard	Arduino UNO
LCD Screen	J15 thru J30	
Potentiometer	D6, D7, D8	
Jumper (RED)	(+) Rail	5 V
Jumper (BLACK)	(-) Rail	GND
Jumper (RED)	(+) Rail	E 6
Jumper (BLACK)	(-) Rail	E 8
Jumper (RED)	(+) Rail	F 16
Jumper (BLACK)	(-) Rail	F 15
Jumper (RED)	(+) Rail	F 29
Jumper (BLACK)	(-) Rail	F 26
Jumper (BLACK)	(-) Rail	F 30
Jumper (ORANGE)	E 7	F 28

Component	Breadboard	Arduino UNO
Jumper (YELLOW)	F 27	Pin 12
Jumper (BLUE)	F 25	Pin 11
Jumper (ORANGE)	F 20	Pin 5
Jumper (YELLOW)	F 17	Pin 2
Jumper (BLUE)	F 18	Pin 3
Jumper (GREEN)	F 19	Pin 4

7

The Arduino Bit From littleBits

The Arduino Bit From littleBits[®]



Another way to get started with Arduino is by using the Arduino bit from littleBits[®]

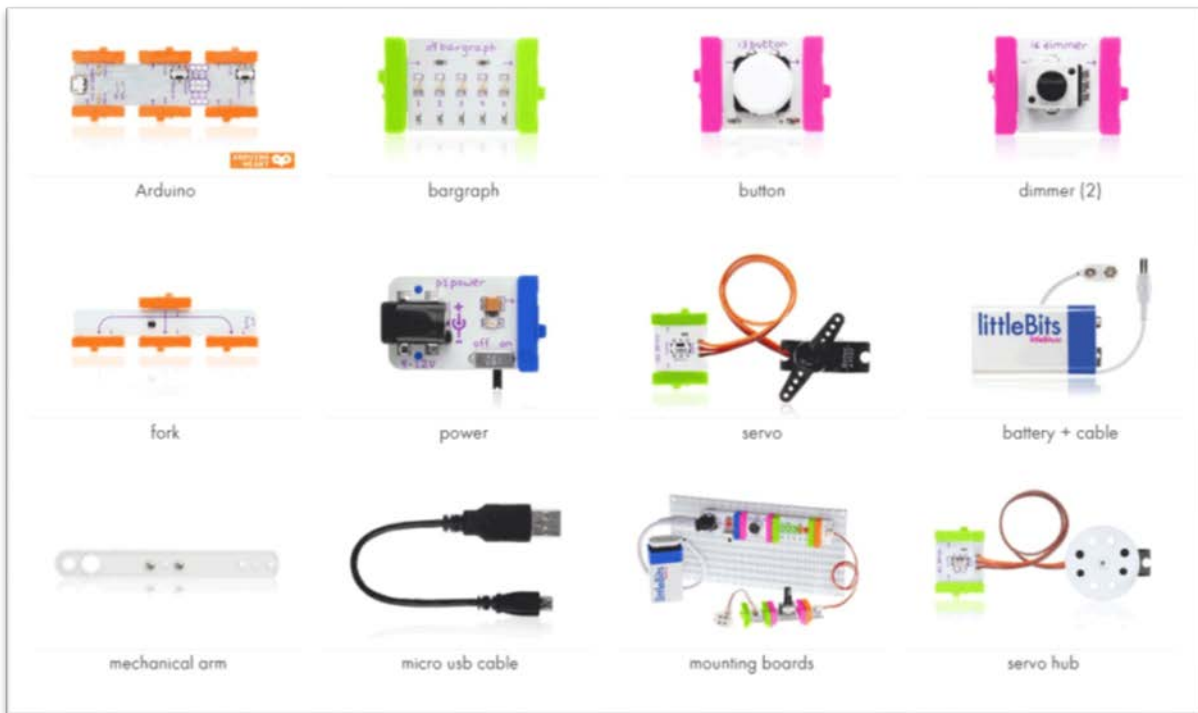
Intro To littleBits

For those of you not familiar, littleBits is a collection of electronic modules known as bits that connect together using magnets. Each bit has a specific function such as an input or an output like an LED or a motor. Snap them together and you can create circuits in seconds with no programming required.

But what if you wanted to create and code your own inventions without having to use the traditional Arduino, breadboard and jumper wire setup? That's where the Arduino Coding Kit comes in and is a perfect fit. This kit includes an Arduino bit and other accessories that will allow you to code your own interactive hardware projects.

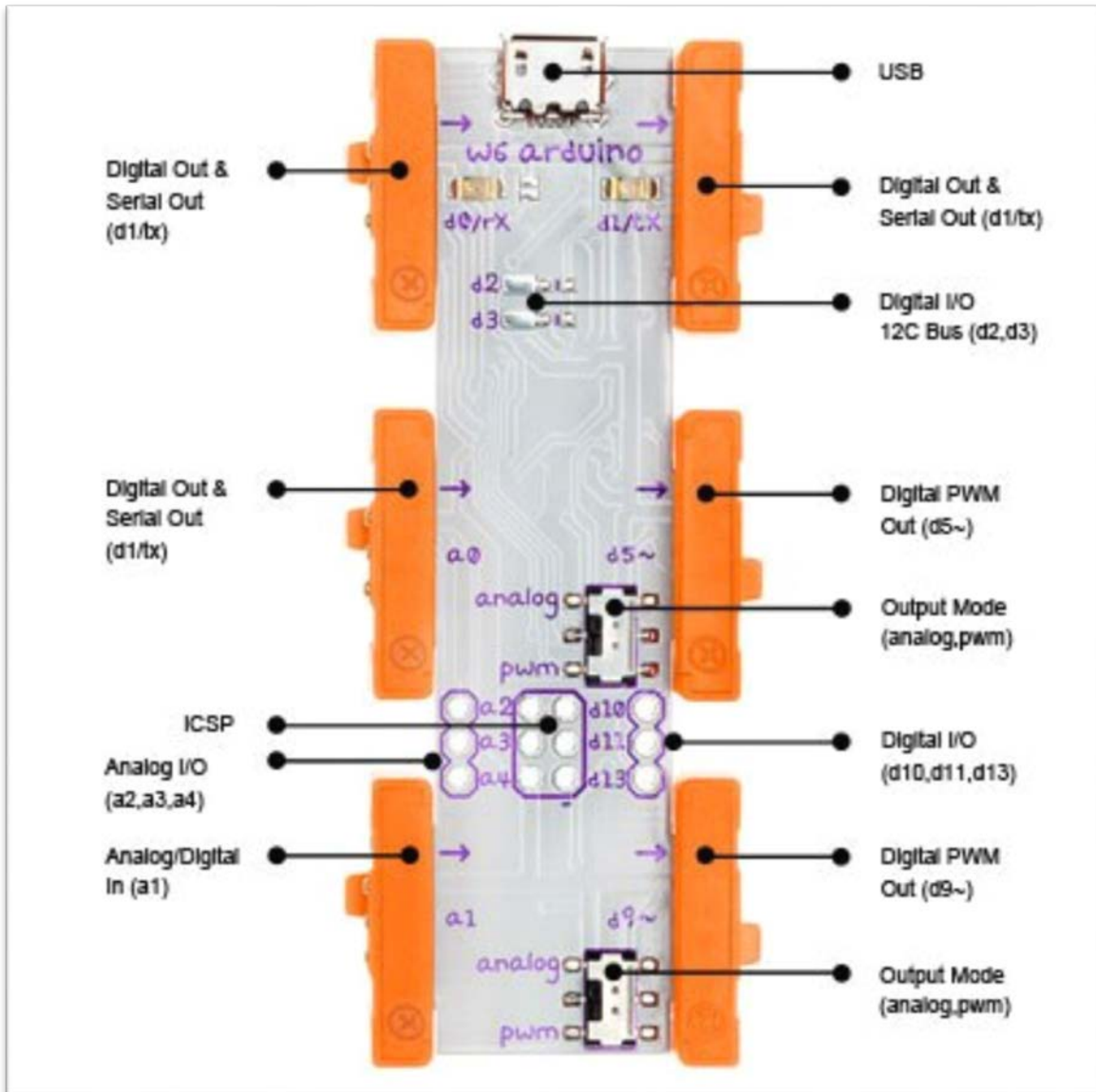
littleBits Arduino Coding Kit

The following are the bits and accessories that are included in the littleBits Arduino Coding Kit. For the following two projects, you will only need a few of them and they could be purchased separately if needed.



Arduino Bit Breakdown

The “brains” of the Coding Kit is the Arduino bit and the below graphic helps to point out it’s key components. If you want more information on this bit, check out this [Arduino bit tutorial](#) from littleBits.



Project 1- Blink an LED

Now it's time to blink an LED using an Arduino bit from littleBits®.

Parts Required

Arduino Bit

Power Bit

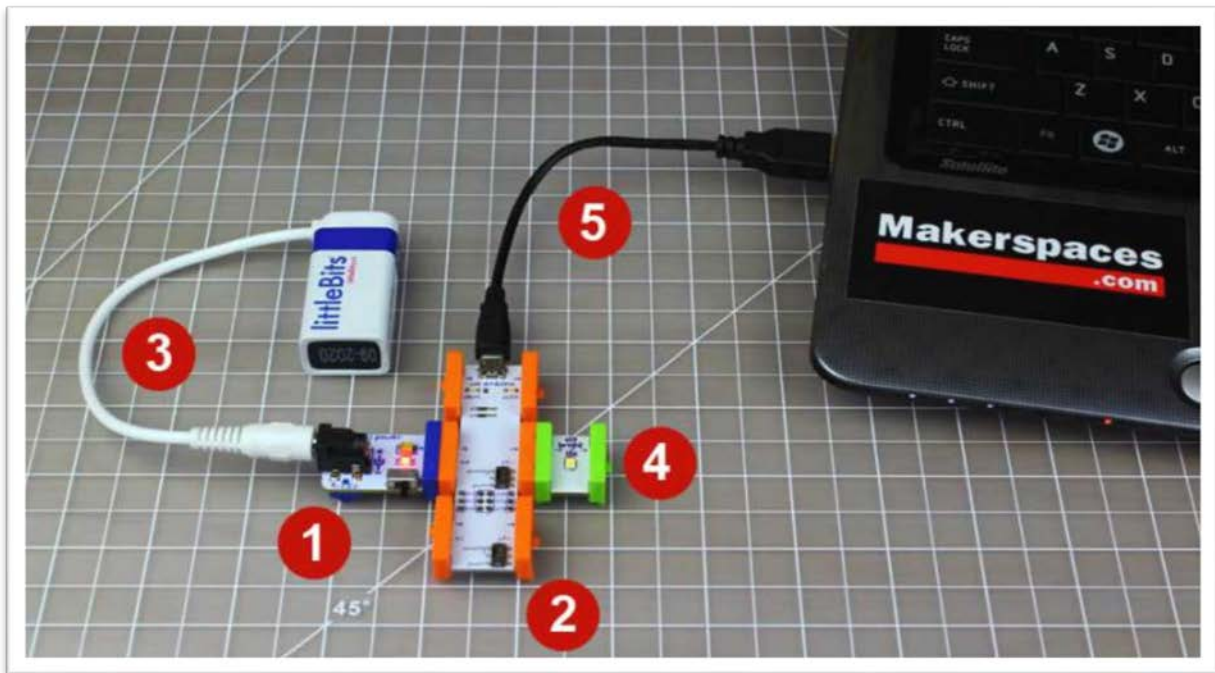
Battery & Power Cable

Micro USB Cable

Bright LED Bit

Step 1 – Create LED Blink Circuit

To begin, you will need to connect the power bit **(1)** to the Arduino bit **(2)** at the point marked (a- zero). Make sure the 9v battery **(3)** is attached to the power bit using the power cable. Next, attach the LED bit **(4)** to the side marked (d5~) and then **TURN ON** the power bit. The last step is to connect the Arduino bit to your computer **(5)** using the micro USB cable.

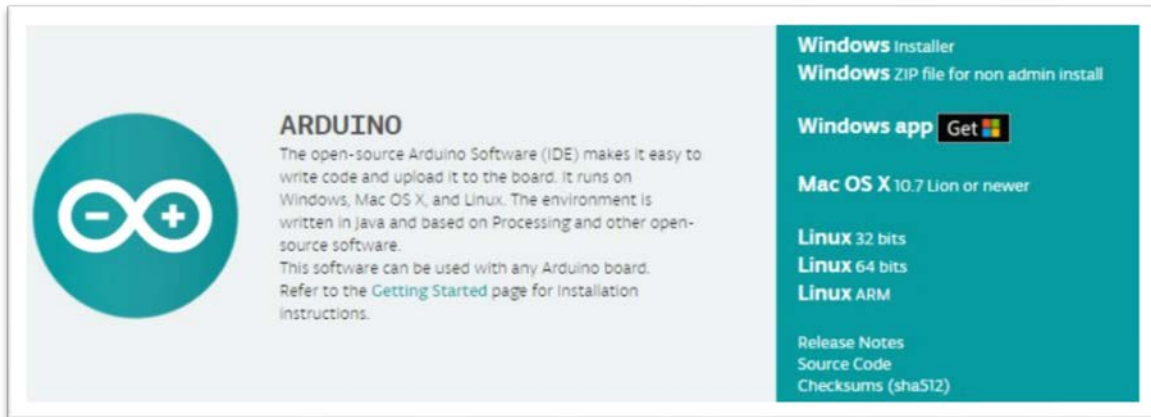


Step 2 – Download Arduino IDE Software

The Arduino IDE is free software that is used to write the code that is uploaded to the littleBits Arduino bit.

To download the software, use the following link – www.arduino.cc/en/Main/Software

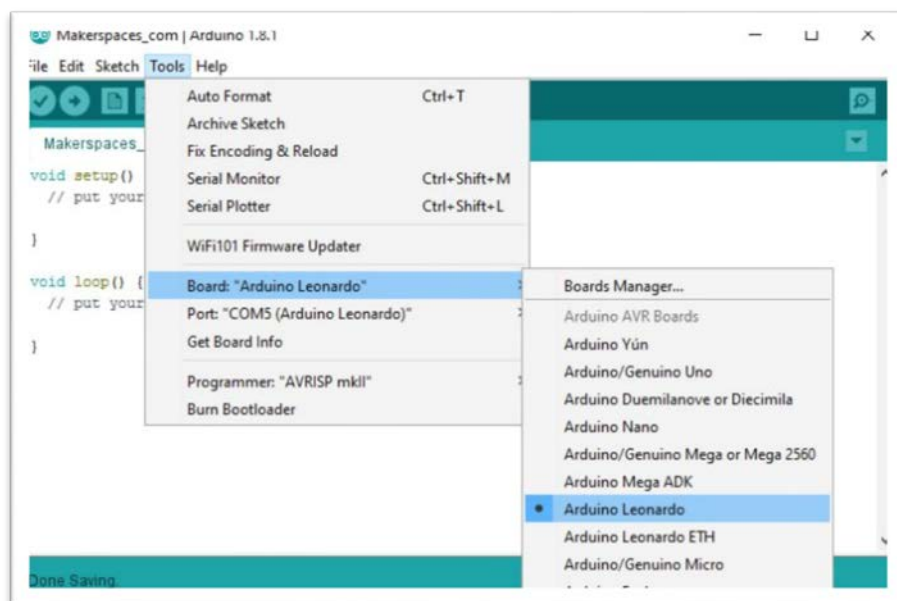
Select the operating system you have and click on the appropriate link on the right-hand side.



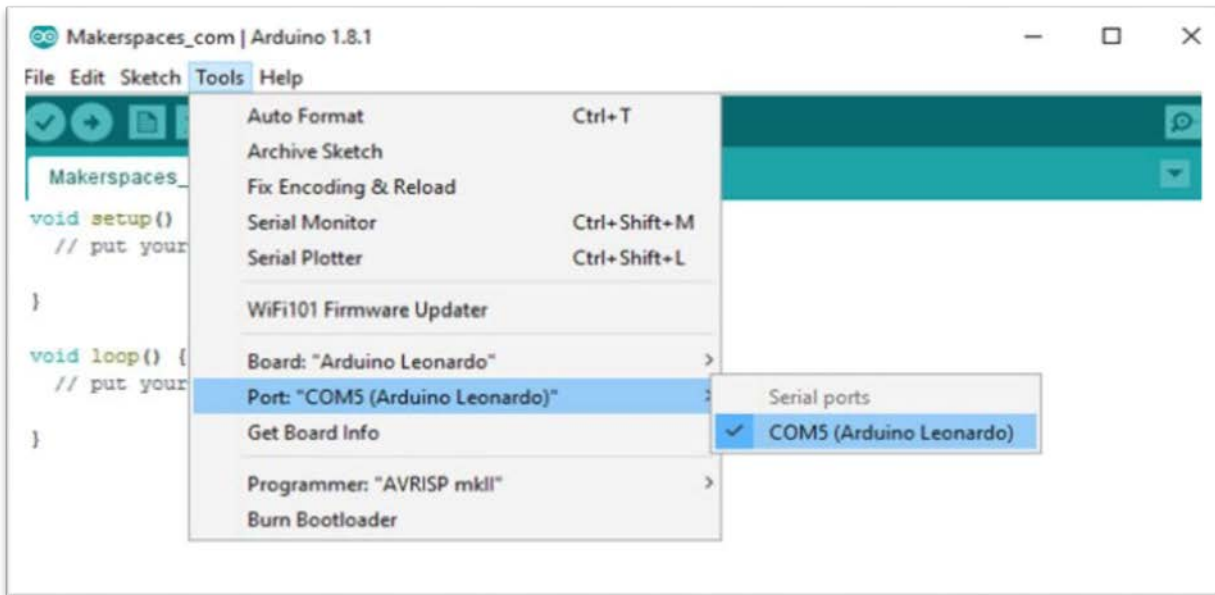
Step 3 – Select The littleBits Arduino Board & Port

Open up the IDE software and make sure the Arduino bit is connected to your computer and the power bit is switched on.

Select **Tools > Board > Arduino Leonardo**



Next you need to select the port the Arduino will be connected to. Go to **Tools > Port > ComX (Arduino Leonardo)**



Step 4 – Upload Blink Sketch To Arduino Bit

Now it's time to upload the code (known as a sketch) to the Arduino bit. First, download the below zip file and click extract.

Blink Sketch – Zip File

Open the "Blink" file and click the forward arrow in the top left of the IDE. At this point, the code is being uploaded and the LED bit should start blinking in a few seconds. Try experimenting with the code and change the **delay** portion from (200) to (1000). What does it do?

Note – In order for the "Blink" file to work correctly, it will need to be located in a folder named Blink.

Project 2- Fade a Bargraph

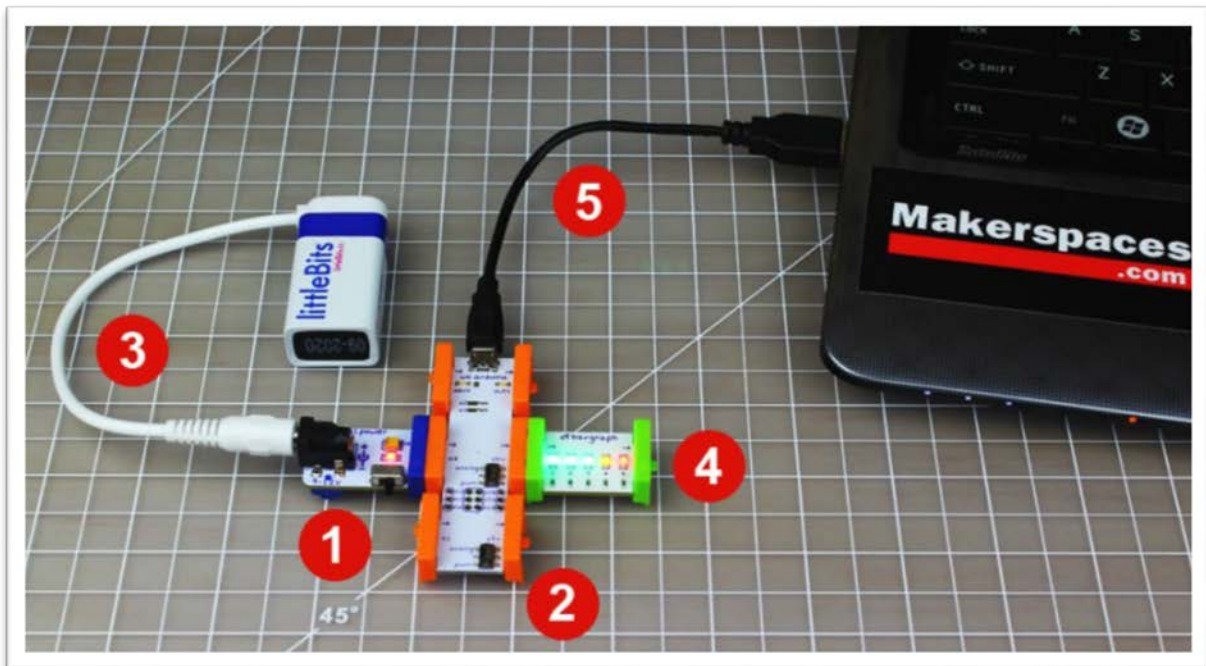
Now it's time to create a fading LED bargraph using the littleBits Arduino bit.

Parts Required

- [Arduino Bit](#)
- [Power Bit](#)
- [Battery & Power Cable](#)
- [Micro USB Cable](#)
- [Bargraph Bit](#)

Step 1 – Connect The Bits

To begin, you will need to connect the power bit **(1)** to the Arduino bit **(2)** at the point marked (a- zero). Make sure the 9v battery **(3)** is attached to the power bit using the power cable. Next, attach the bargraph bit **(4)** to the side marked (d5~) and then **TURN ON** the power bit. The last step is to connect the Arduino bit to your computer **(5)** using the micro USB cable.



Step 2 – Upload Fade Sketch To Arduino Bit

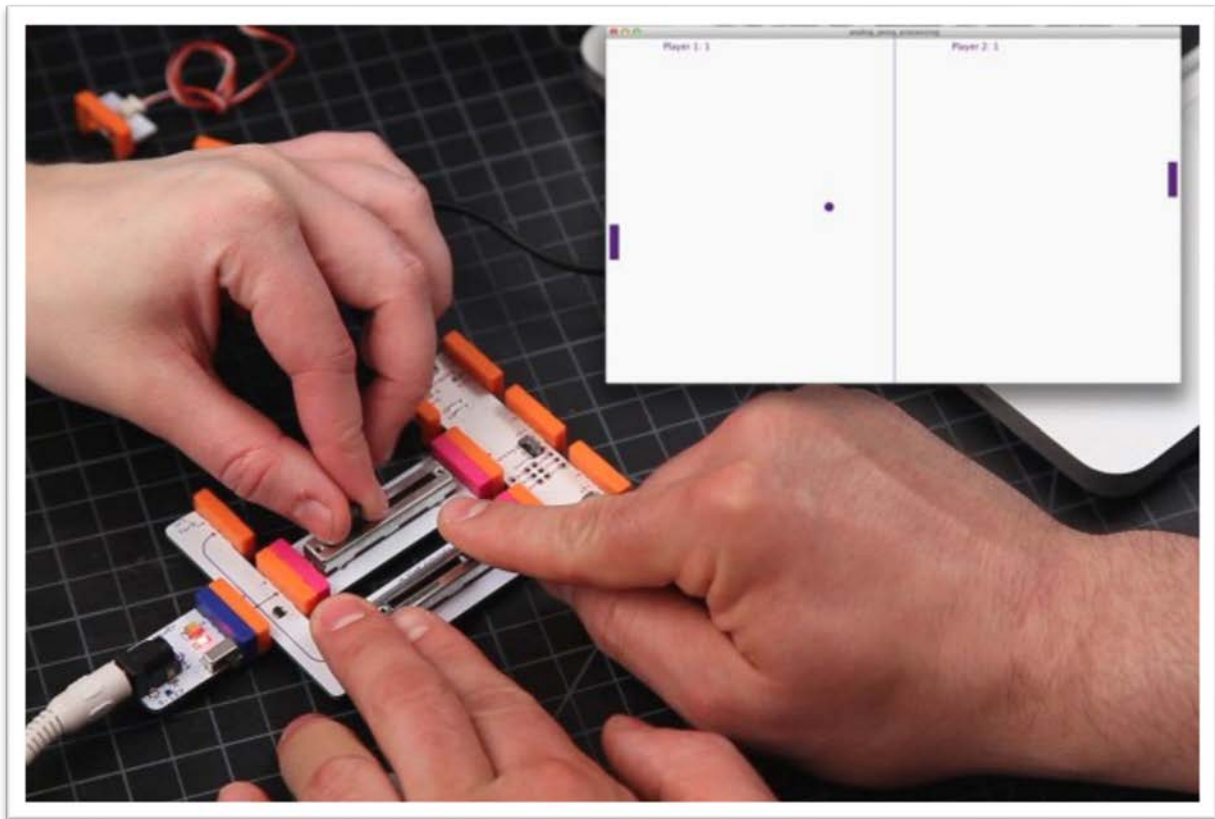
Download the below zip file and click extract.

Fade Sketch – Zip File

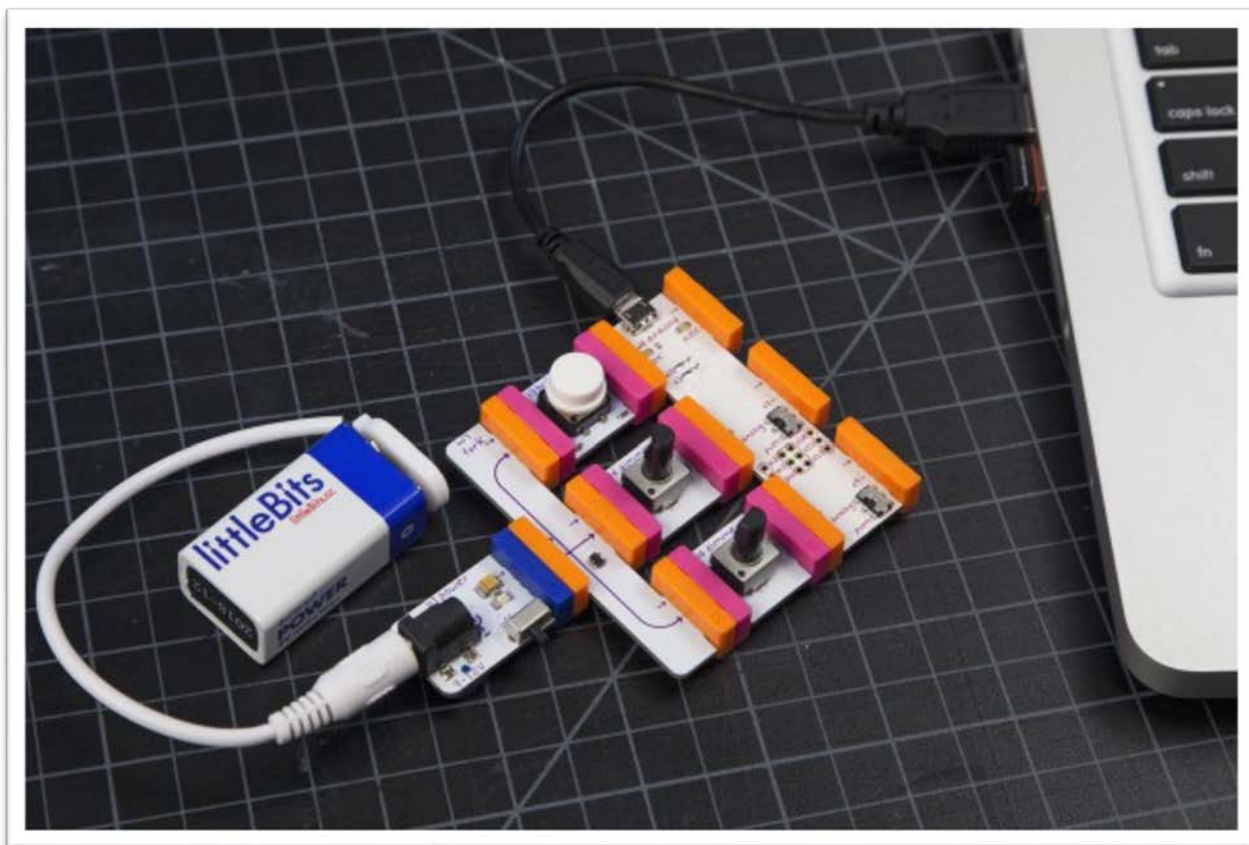
Open the “Fade” file and click the forward arrow in the top left of the IDE. At this point, the code is being uploaded to the bit and the bargraph should start fading.

More Arduino Projects

Looking for more beginner Arduino projects that involve littleBits? Check out their [getting started section](#) for more ideas. Below are a few examples of what you’ll find.



LittleBits Analog Pong Game



littleBits DIY Computer Mouse

8

Arduino Trouble - Shooting

Arduino Troubleshooting

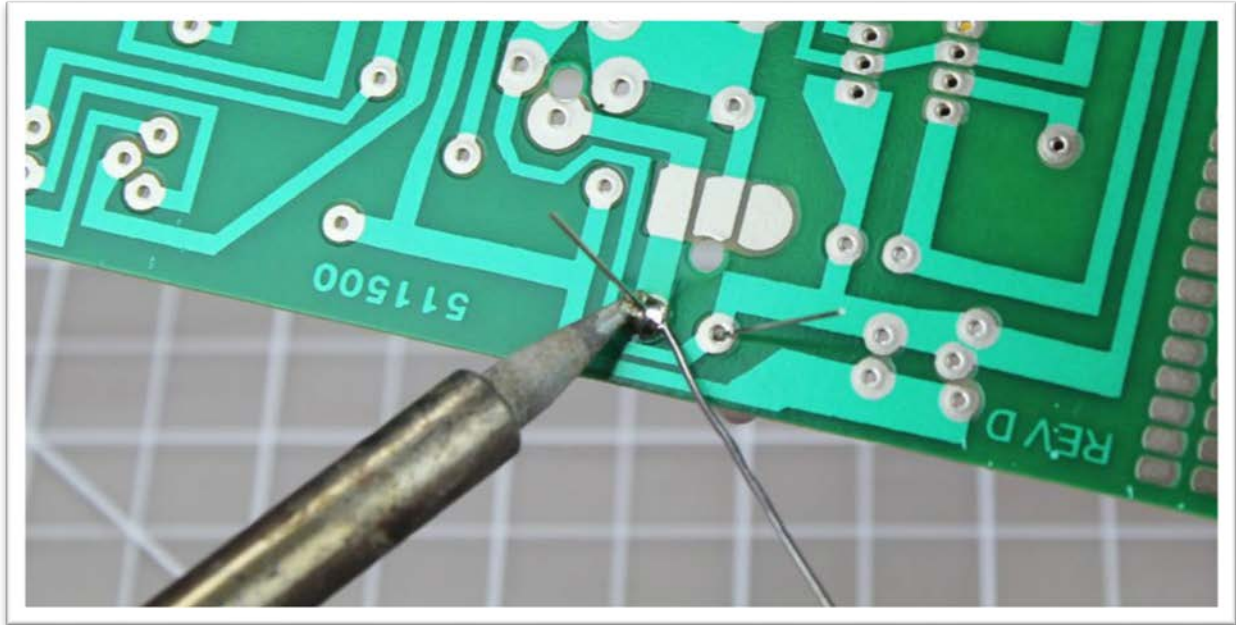
- Make sure the Arduino IDE shows the correct board. Go to **Tools** > **Board** then select **Arduino Uno**.
- Make sure the Arduino IDE shows the correct port. Go to **Tools** > **Port** then select the port that says **Arduino**.
- It can be easy to put a component or jumper into the wrong pin on the Arduino or the breadboard. Double check the correct pin is being used.
- Verify all component connections are secure with the Arduino board and breadboard.
- Verify the LED is actually functional. Use a 3v coin cell battery and connect the LONG leg of the LED to the (+) and SHORT leg to the (-) of the battery.
- The long leg of the LED is the (+) positive and the short leg is the (-) negative. Make sure the correct leg of the LED is in the proper pin of the Arduino or breadboard as directed.

9

How To Solder Your Arduino Projects

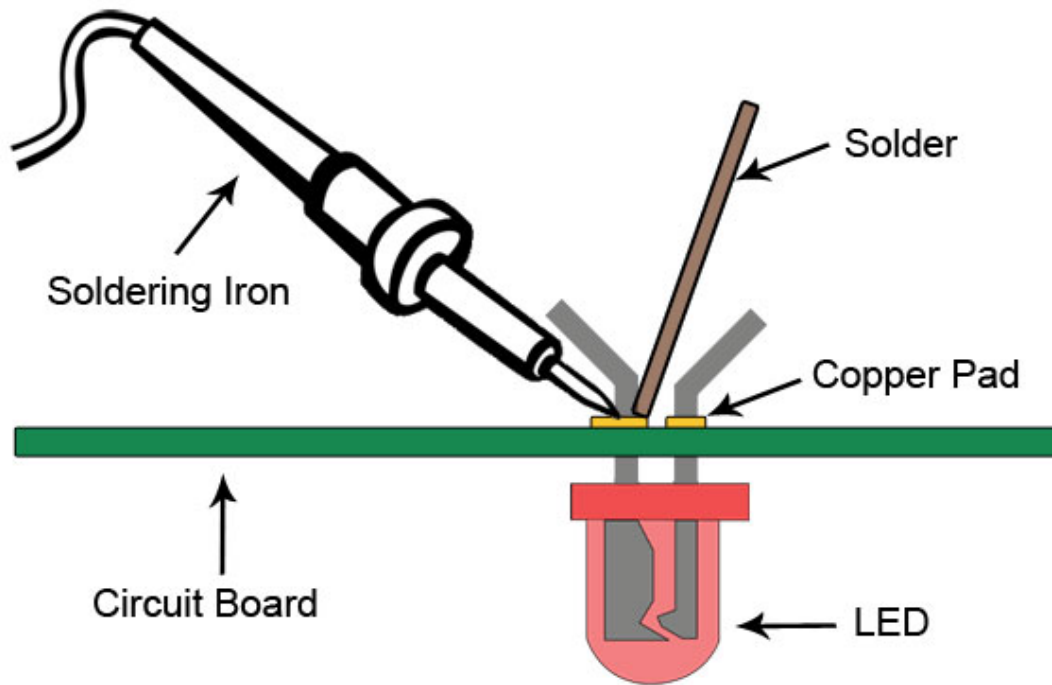
How To Solder Your Arduino Projects

Learning how to solder w/ proper soldering techniques is a fundamental skill every maker should master. This skill also comes in handy when you are ready to make your Arduino creations more permanent. In this chapter, we'll outline the basics of soldering and show you what tools you'll need to do it correctly.



What Is Soldering?

If you were to take apart any electronic device that contains a circuit board, you'll see the components are attached using soldering techniques. Soldering is the process of joining two or more electronic parts together by melting solder around the connection. Solder is a metal alloy and when it cools it creates a strong electrical bond between the parts. Even though soldering can create a permanent connection, it can also be reversed using a desoldering tool.



Soldering Tools & Materials

The good thing about learning how to solder is the fact that you don't need a lot to get started. Below we'll outline the basic tools and materials you will need for most of your soldering projects.

Soldering Iron

A soldering iron is a hand tool that plugs into a standard 120v AC outlet and heats up in order to melt solder around electrical connections. This is one of the most important tools used in soldering and it can come in a few variations such as pen or gun form. For beginners, it's recommended that you use the pen style soldering iron in the 15W to 30W range. Most soldering irons have interchangeable tips that can be used for different soldering applications. Be very cautious when using any type of soldering iron because it can heat up to 896° F which is extremely hot.



Soldering Station

A soldering station is a more advanced version of the basic standalone soldering pen. If you are going to be doing a lot of soldering, these are great to have as they offer more flexibility and control. The main benefit of a soldering station is the ability to precisely adjust the temperature of the soldering iron which is great for a range of projects. These stations can also create a safer workspace as some include advanced temperature sensors, alert settings and even password protection for safety.



Soldering Iron Tips

At the end of most soldering irons is an interchangeable part known as a soldering tip. There are many variations of this tip and they come in a wide variety of shapes and sizes. Each tip is used for a specific purpose and offers a distinct advantage over another. The most common tips you will use in electronics projects are the conical tip and the chisel tip.

Conical Tip – Used in precision electronics soldering because of the fine tip. Because of its pointed end, it's able to deliver heat to smaller areas without affecting its surroundings.

Chisel Tip – This tip is well-suited to soldering wires or other larger components because of its broad flat tip.



Image Credit – Sparkfun.com

Brass or Conventional Sponge

Using a sponge will help to keep the soldering iron tip clean by removing the oxidation that forms. Tips with oxidation will tend to turn black and not accept solder as it did when it was new. You could use a conventional wet sponge but this tends to shorten the lifespan of the tip due to expansion and contraction. Also, a wet sponge will drop the temperature of the tip temporarily when wiped. A better alternative is to use a brass sponge as shown on the left.



Soldering Iron Stand

A soldering iron stand is very basic but very useful and handy to have. This stand helps prevent the hot iron tip from coming in contact with flammable materials or causing accidental injury to your hand. Most soldering stations come with this built in and also include a sponge or brass sponge for cleaning the tip.



Solder

Solder is a metal alloy material that is melted to create a permanent bond between electrical parts. It comes in both lead and lead-free variations with diameters of .032" and .062" being the most common. Inside the solder core is a material known as flux which helps improve electrical contact and its mechanical strength.

For electronics soldering, the most commonly used type is lead-free rosin core solder. This type of solder is usually made up of a Tin/Copper alloy. You can also use leaded 60/40 (60% tin, 40% lead) rosin core solder but it's becoming less popular due to health concerns. If you do use lead solder, make sure you have proper ventilation and that you wash your hands after use.



When buying solder, make sure NOT to use acid core solder as this will damage your circuits and components. Acid core solder is sold at home improvement stores and is mainly used for plumbing and metal working.

As mentioned earlier, solder does come in a few different diameters. The thicker diameter solder (.062") is good for soldering larger joints more quickly but it can make soldering smaller joints difficult. For this reason, it's always a good idea to have both sizes on hand for your different projects.

Helping Hand (3rd Hand)

A helping hand is a device that has 2 or more alligator clips and sometimes a magnifying glass/light attached. These clips will assist you by holding the items you are trying to solder while you use the soldering iron and solder. A 3rd hand is a very helpful tool to have in your makerspace when working with electronics or solder.



Safety

Now that you know what tools and materials are required, it's time to briefly discuss ways of staying safe while soldering.

Soldering irons can reach temperatures of 800° F so it's very important to know where your iron is at all times. We always recommend you use a soldering iron stand to help prevent accidental burns or damage.



Make sure you are soldering in a well ventilated area. When solder is heated, there are fumes released that are harmful to your eyes and lungs. It's recommended to use a fume extractor which is a fan with a charcoal filter that absorbs the harmful solder smoke.

It's always a good idea to wear protective eye wear in case of accidental splashes of hot solder. Lastly, make sure to wash your hands when you are done soldering especially if using lead solder.

How To Solder

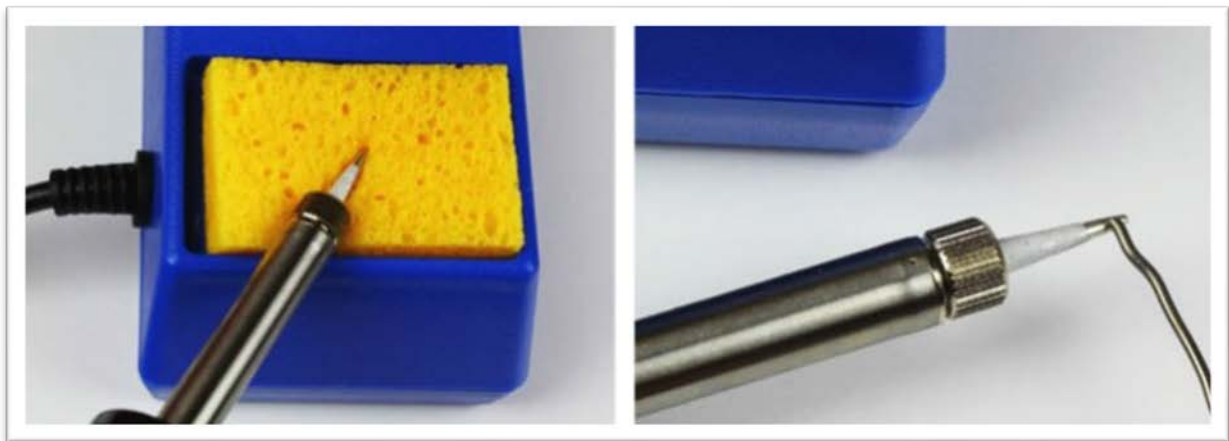
Before you can start soldering, you need to prep your soldering iron by tinning the tip with solder. This process will help improve the heat transfer from the iron to the item you're soldering. Tinning will also help to protect the tip and reduce wear.

Step 1: Begin by making sure the tip is attached to the iron and screwed tightly in place.

Step 2: Turn on your soldering iron and let it heat up. If you have a soldering station with an adjustable temp control, set it to 400° C/ 752° F.

Step 3: Wipe the tip of the soldering iron on a damp wet sponge to clean it. Wait a few seconds to let the tip heat up again before proceeding to step 4.

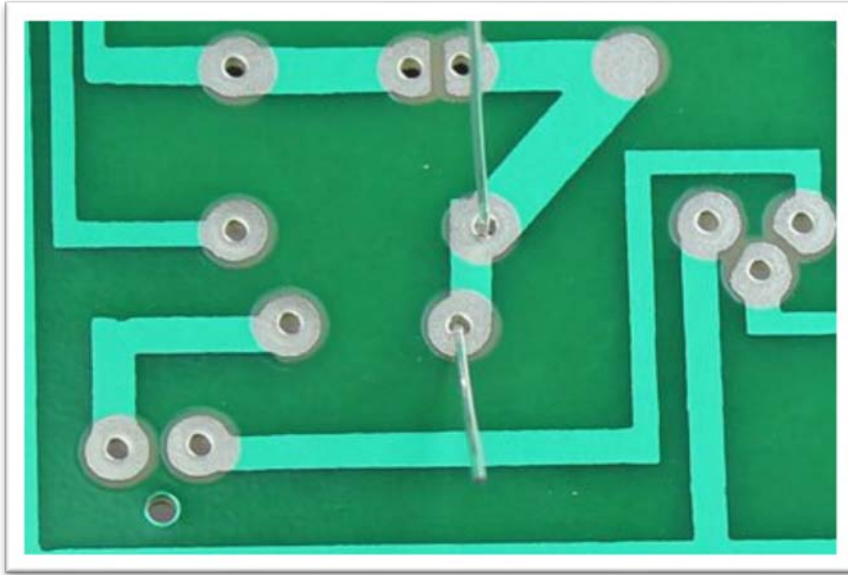
Step 4: Hold the soldering iron in one hand and solder in the other. Touch the solder to the tip of the iron and make sure the solder flows evenly around the tip.



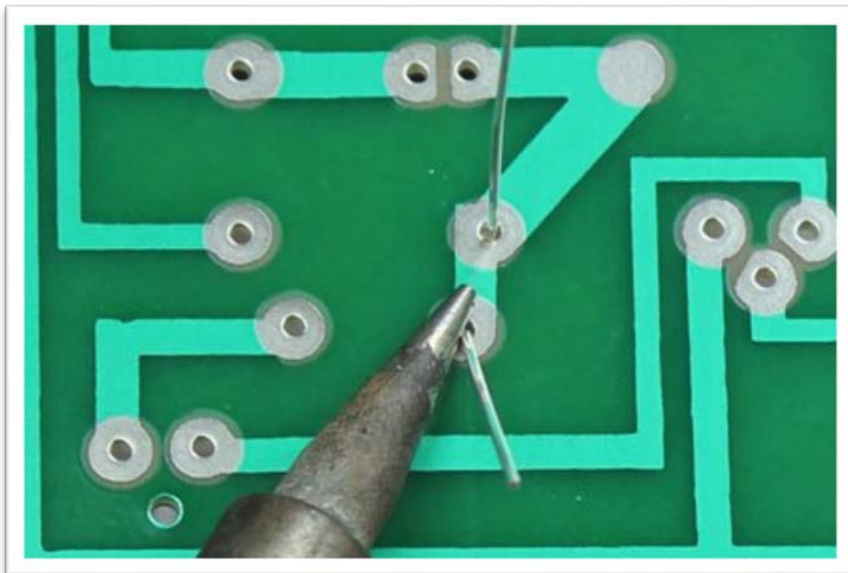
You should tin the tip of your iron before and after each soldering session to extend its life. Eventually, every tip will wear out and will need replacing when it becomes rough or pitted.

Now that the soldering iron tip has been prepped, we're ready to start soldering electronics. In this example, we'll be soldering an LED to a circuit board.

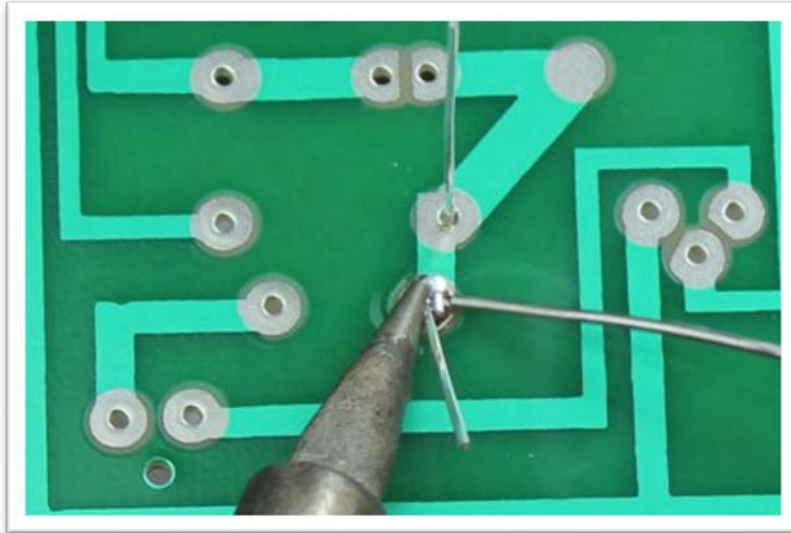
Step 1: Mount The Component – Begin by inserting the leads of the LED into the holes of the circuit board. Flip the board over and bend the leads outward at a 45° angle. This will help the component make a better connection with the copper pad and prevent it from falling out while soldering.



Step 2: Heat The Joint – Turn your soldering iron on and if it has an adjustable heat control, set it to 400°C. At this point, touch the tip of the iron to the copper pad and the resistor lead at the same time. You need to hold the soldering iron in place for 3-4 seconds in order to heat the pad and the lead.

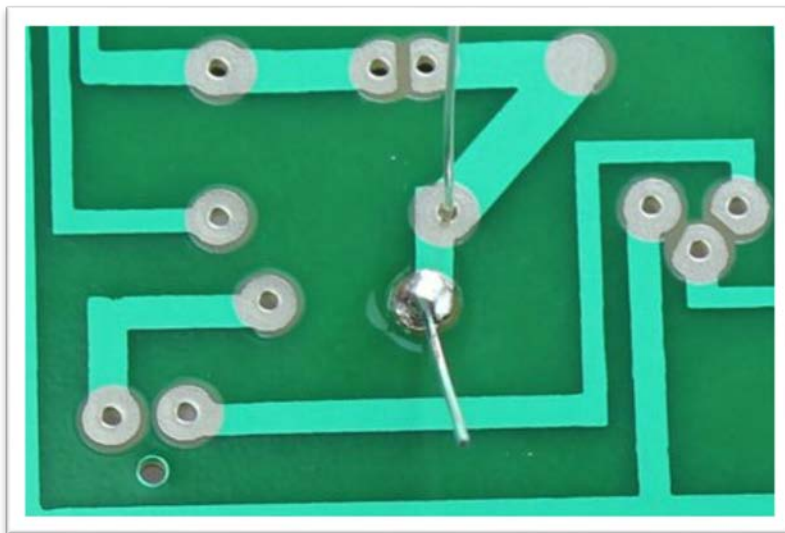


Step 3: Apply Solder To Joint – Continue holding the soldering iron on the copper pad and the lead and touch your solder to the joint. **IMPORTANT** – Don't touch the solder directly to the tip of the iron. You want the joint to be hot enough to melt the solder when it's touched. If the joint is too cold, it will form a bad connection.



Step 4: Snip The Leads – Remove the soldering iron and let the solder cool down naturally. Don't blow on the solder as this will cause a bad joint. Once cool, you can snip the extra wire from leads.

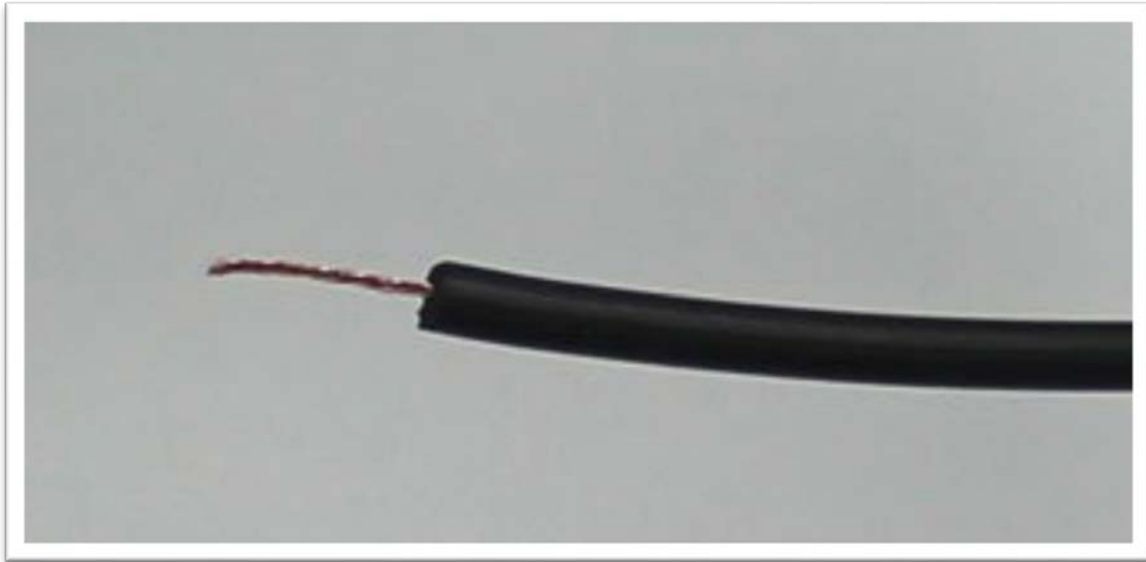
A proper solder joint is smooth, shiny and looks like a volcano or cone shape. You want just enough solder to cover the entire joint but not too much so it becomes a ball or spills to a nearby lead or joint.



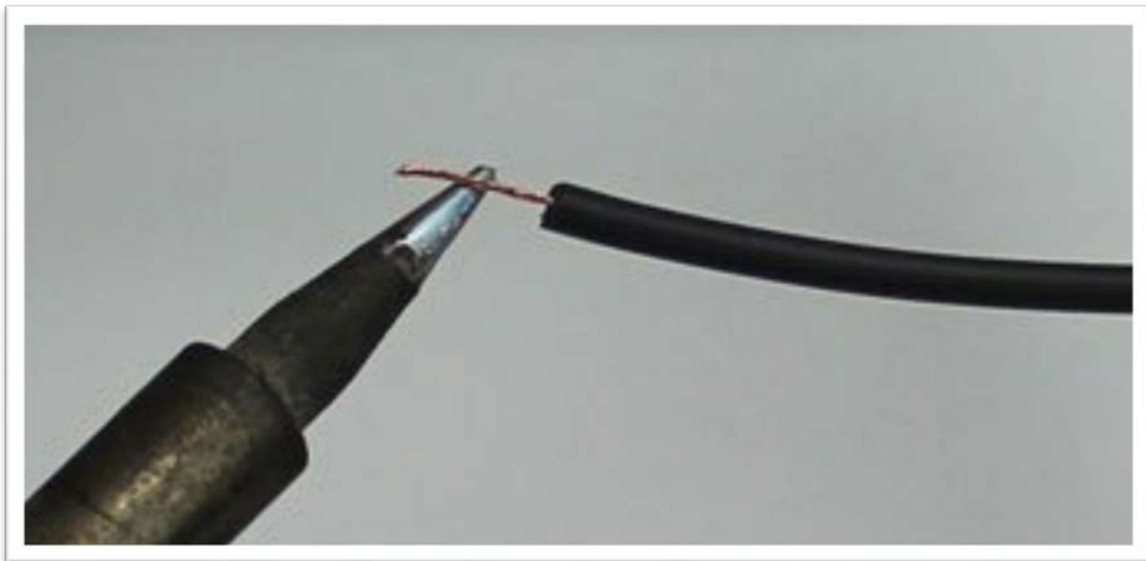
How To Solder Wires

Now it's time to show you how to solder wires together. For this process, it's recommended to use helping hands or other type of clamp device.

Begin by removing the insulation from the ends of both wires you are soldering together. If the wire is stranded, twist the strands together with your fingers.



Make sure your soldering iron is fully heated and touch the tip to the end of one of the wires. Hold it on the wire for 3-4 seconds.



Keep the iron in place and touch the solder to the wire until it's fully coated. Repeat this process on the other wire.



Hold the two tinned wires on top of each other and touch the soldering iron to both wires. This process should melt the solder and coat both wires evenly.



Remove the soldering iron and wait a few seconds to let the soldered connection cool and harden. Use heat shrink to cover the exposed connection.



How To Desolder

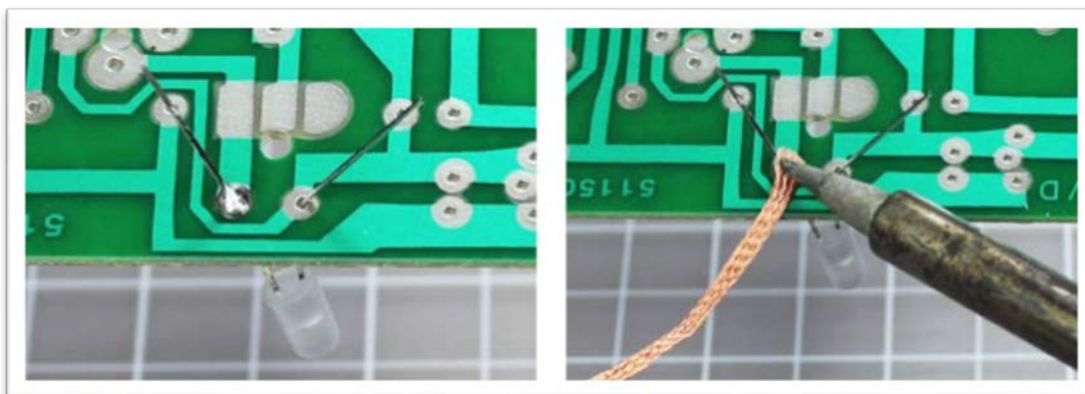
The good thing about using solder is the fact that it can be removed easily in a technique known as desoldering. This comes in handy if you need to remove a component or make a correction to your electronic circuit.

To desolder a joint, you will need solder wick which is also known as desoldering braid.



Step 1 – Place a piece of the desoldering braid on top of the joint/solder you want removed.

Step 2 – Heat your soldering iron and touch the tip to the top of the braid. This will heat the solder below which will then be absorbed into the desoldering braid. You can now remove the braid to see the solder has been extracted and removed. Be careful touching the braid when you are heating it because it will get hot.



Optional – If you have a lot of solder you want removed, you may want to use a device called a solder sucker. This is a handheld mechanical vacuum that sucks up hot solder with a press of a button.

To use, press the plunger down at the end of the solder sucker. Heat the joint with your soldering iron and place the tip of the solder sucker over the hot solder. Press the release button to suck up the liquid solder. In order to empty the solder sucker, press down on the plunger.



10

**Arduino &
Electronics
Suppliers**

Arduino & Electronics Suppliers

There are a lot of great places online to find Arduino & electronic components, parts and tools. Below is a list of some of our favorite places to go.

Adafruit	adafruit.com
Sparkfun	sparkfun.com
Jameco	jameco.com
Mouser	mouser.com
Velleman Store	vellemanstore.com
DigiKey	digikey.com
All Electronics	allelectronics.com
MCM Electronics	mcmelectronics.com
MakerShed	makershed.com

11

**Never
Stop
Learning**

Never Stop Learning

We're always online sharing info, projects and ideas regarding maker education and makerspaces. Below are some of the places you can find us. Make sure to stay in touch and show us what you're working on or what questions you have.

Web	www.makerspaces.com
Twitter	@makerspaces.com @andrewbmiller
Facebook	www.fb.com/makerspaces
Instagram	www.instagram.com/makerspaces
Pinterest	www.pinterest.com/maker_spaces