

The Arduino is a fun and stress-free introduction into the world of coding and engineering that allows you to use code to control devices, such as LEDs and Servos. We recommend you read through this document for an introduction to the Arduino and breadboards. As other devices are introduced in each activity, they will be explained. Below is a recommended order of steps to prepare for the Arduino activities and the subsequent pages will provide detailed instructions for each step. Whether completing the Arduino activities on your own or leading a group of people through the activities, we recommend that you start here.

Step 1: Download the Arduino Software

While this step may be quick for some, others have reported that this held them up the longest. *Most importantly, download the Arduino software to a computer, not a tablet.* The connection between the Arduino program and the Arduino board is through a USB port, therefore ensure computers have a usable USB port.

Step 2: Purchase Your Supplies

Order a complete kit, or purchase items individually and make the kit yourself. This section helps you decide which is the best option.

Step 3: Learn About the Arduino Uno

The Arduino Uno is a simple device, but can look complicating and doesn't typically come with instruction. This section will guide you through learning the important parts of the Arduino Uno.

Step 4: Learn About Breadboards

While breadboards are useful and commonly used tools, students often do not understand what breadboards are or how they work. This section explains the breadboards and identifies the differences between standard and mini breadboards.

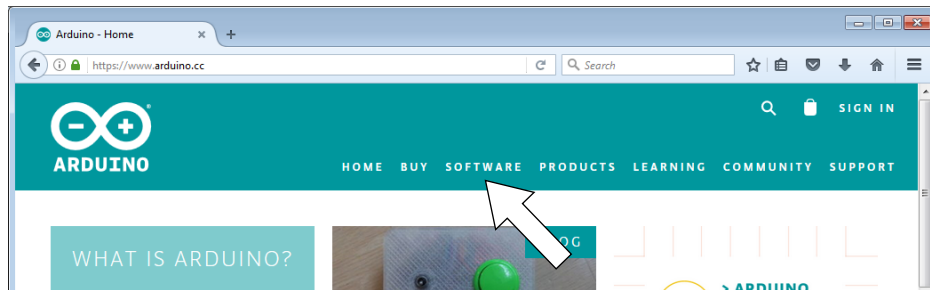
Once you have completed the 4 steps above, use the Arduino Uno Activities to learn and teach, which can be found at <https://www.thinkabitlab.com/arduino-activities>. The Arduino Uno Activities are divided into different content areas (LEDs, Servos, and Sensors), but often rely on previous activities. We recommend that you read steps explaining what code to write and wires to connect, in addition to the extra explanations that go along with each step. This valuable information provides a better understanding of the Arduino software and empowers you to do more activities individually.

These activities are to be implemented by a responsible adult instructor who has a basic understanding of safety considerations (both in general as well as of the specific safety considerations identified herein) when implementing hands-on activities that can potentially cause harm to students/participants, if not implemented properly. In addition to following all general safety precautions, adults must review and adhere to all indicated safety precautions when implementing the activity.

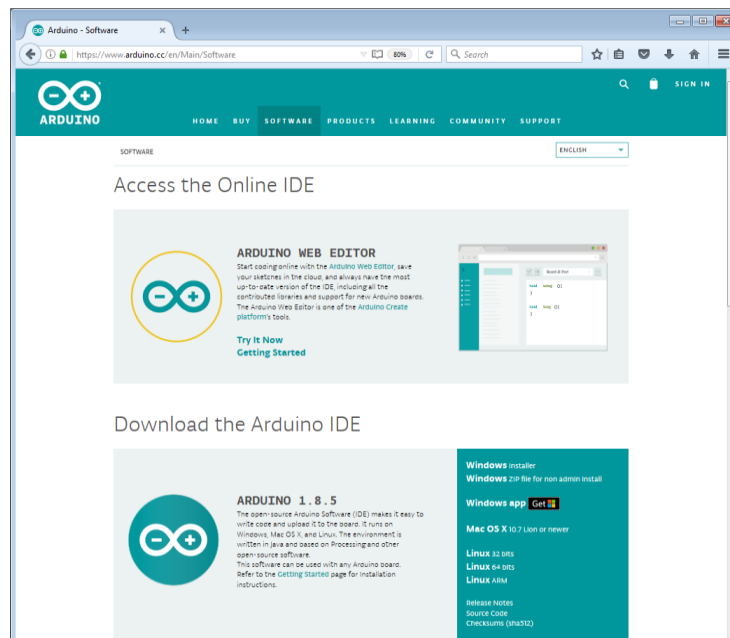
Never allow the red and black wires to touch together while they are connected to a powered Arduino board as that creates a short circuit. A short circuit can potentially cause the wires and/or the Arduino board to get hot enough to burn the skin. In addition, in the event of a short circuit the Arduino board can potentially catch on fire.

This section explains how to download and install the Arduino software. The software is needed when using your computer to write programs for the Arduino board.

1. First, go to www.arduino.cc and click on "SOFTWARE" tab.



2. Next, decide on the best IDE: (I) Access the Online IDE, i.e., Arduino Web Editor, or (II) Download the Arduino IDE. The Web Editor permits the user to create an online Arduino account and saves sketches in the cloud to access them anywhere (with internet connectivity). For the software downloaded to the desktop, no internet connectivity is needed. However, please note that there are different downloads for different operating systems, be sure to select the link that matches the computer in use.



3. Then, use the Arduino IDE to write code to send to Arduino. Once a code is sent to the Arduino, it lives on the Arduino Uno. Any future edits to that code on the computer will not be sent to the Arduino unless it is uploaded to the Arduino Uno. When using the Arduino software with a large class, we recommend testing the Arduino IDE by completing the LED Blink activity on each computer.

Arduino Uno Starter Kits

This section explains the Arduino Uno Starter Kit. The starter kits contain all the components needed to build a few simple but fun projects that will teach the basics of circuits and Arduino programming. These kits are one of the best ways to get started with Arduino, but there are many different starter kits to choose from, so research before buying.

The Arduino website (www.arduino.cc) sells Arduino Uno Starter Kit, but Amazon also sells several different starter kits sometimes at a lower price. Most starter kits include a guidebook that explains how to wire and code different projects and introduces multiple activities. Complete the activities in the starter kit to gain the skills needed to do other projects. The kit contains most of the components used in the listed activities, along with many other parts. Note that the full rotation Servo is not included in the starter kit and can be bought separately for about \$12.

The guidebook that comes with the official Arduino Starter Kit is very helpful. It shows a circuit schematic for each activity and includes a wiring diagram on how to connect all the components. Also, the code for each operation comes pre-loaded with the Arduino software, and the guidebook describes an explanation for each line of code. Additionally, the book implies how to apply the skills learned during each activity to other projects. Lastly, the components included with the official Arduino Starter Kit are fantastic. The kit package consists of an LCD screen, several types of sensors and buttons, a small Servo, a DC motor, LEDs, resistors, and more. These parts are everything needed to make the projects outlined in the book and almost any other imaginable.

Purchasing Materials Separately

The following list provides information for purchasing items separately to create a starter kit (that is used in the LED and Servo activities). For the Sensor activities, additional supplies should be purchased. (Note: links and prices may not reflect current stock and pricing):

Item	Amount needed	Where to Buy	Approx. Price
Arduino	1 per kit	Arduino , Sparkfun , or Amazon	\$25 - \$50
Standard Breadboard	1 per kit	Amazon or Sparkfun	\$5 - \$10
Standard Servo (micro)	1 per kit	Sparkfun	\$8 - \$10
Continuous Servo (micro)	1 per kit	Sparkfun	\$10 - \$12
LEDs and Resistors	1 per kit	Amazon	\$3 for 80 pcs
Box (optional)	1 per kit	Amazon or Staples	\$5 - \$10
USB Cable	1 per kit	Amazon	\$5 - \$10
Red Wire*	1 per class	Amazon	\$10 - \$15
Black Wire*	1 per class	Amazon	\$10 - \$15
White Wire*	1 per class	Amazon	\$10 - \$15
Wire Strippers**	1 per class	Amazon	\$10 - \$15

*It is recommended to purchase spools of wire. Wire should be 22-gauge solid (not stranded).

**Wire strippers are included so that wire can be cut into smaller pieces and stripped at both ends.

What is the Arduino Uno and Why do we Use it?

This section provides an overview of an Arduino Uno and its different parts, in addition to the fundamental procedure of how to use it correctly.

The Arduino Uno is a microcontroller, or mini-computer, capable of controlling a variety of devices such as LEDs, buzzers, and motors. We use Arduino in the Qualcomm® Thinkabit Lab™ because it is easier to use than other microcontrollers, and it is extremely well documented on the internet. Having proper documentation makes the process of getting help simpler while learning how to use it. Another reason we like the Arduino is that it is scalable to the abilities of the user. It can be very basic, but also very advanced. It can do something as simple as making an LED blink, and as sophisticated as operating a 3D printer. Furthermore, sensors or a Bluetooth module can also be attached to it, etc. ... the possibilities with Arduino Uno are endless!

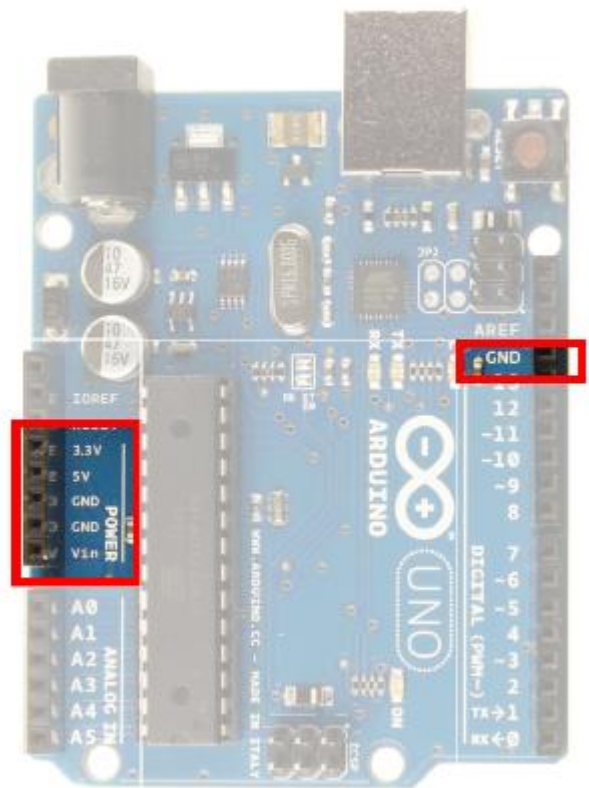
The important components of the Arduino Uno are: (1) The Power Rail, (2) Digital (PWM~), (3) Analog in, and (4) The power supply.



(1) THE POWER RAIL

The power rail is how we get power from the Arduino to the rest of our devices.

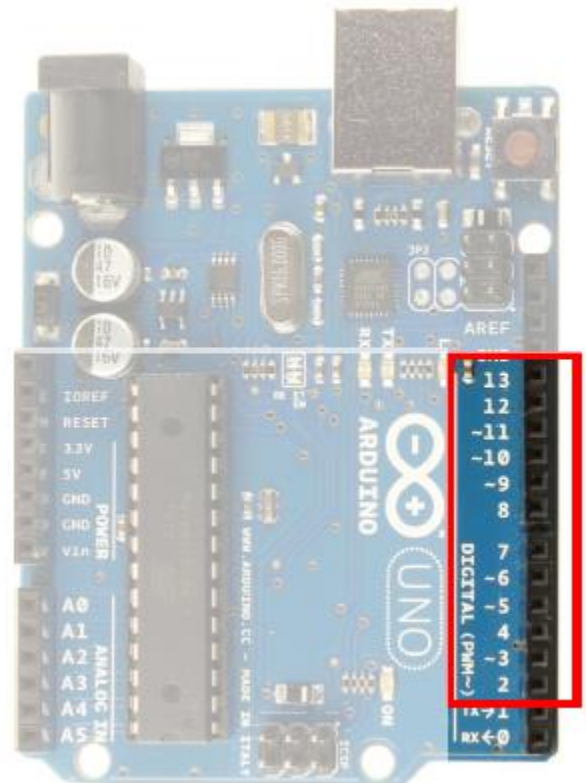
- 3.3V – This pin provides 3.3 volts. This is the “positive side” of power.
- 5V – This pin provides 5 volts. This is also the “positive side” of power.
- GND – This pin, ground, is the “negative side” of power. (Note: there are 3 GND pins on the board and it does not matter which one you use, they’re all connected.)
- Vin – This pin provides the voltage supplied to the Arduino. So, if a 9-volt battery is connected, it will provide 9 volts. This is also the “positive side” of power.
- Be careful not to give attached devices, such as LED’s or Servos, too much power, especially when using the Vin pin.
- Remember if a wire attached to a positive pin touches a wire attached to a negative pin, it will create a “short circuit” and burn out the Arduino.



(2) DIGITAL (PWM~)

The Arduino has “digital” and “analog” pins to input and output electrical signals. Digital signals are either “On” or “Off” (high voltage or low voltage), while analog signals can vary between 0% and 100% on. All digital pins can input and output a digital signal, and some can output an analog signal.

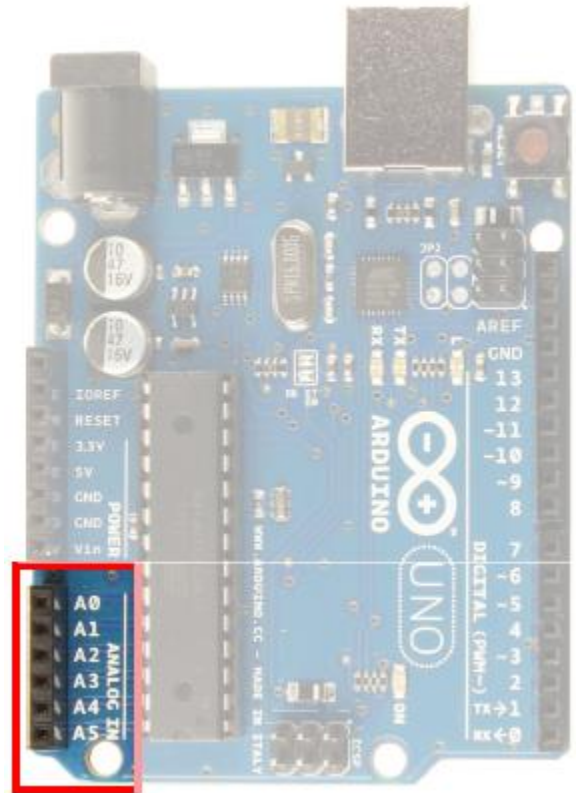
- If a pin’s mode is set to INPUT, then it will sense electrical signals (this is also called “reading”). For example, these digital pins could read the electrical signal generated by pushing a button.
- If a pin’s mode is set to OUTPUT, then it will send out electrical signals (this is also called “writing”). For example, digital pins can be used to send electricity to make an LED turn on.
- Pins with a ~ next to their number can OUTPUT a Pulse Width Modulation (PWM) signal. By turning on and off quickly, they can produce what looks like an analog OUTPUT signal. These are the pins we use to control Servos.
- Notice 0 and 1 are left out. These pins are used for TX (transferring data) and RX (receiving data). We typically don’t use these pins because they are commonly used to troubleshoot the code using the serial monitor.



(3) ANALOG IN

The analog pins on the Arduino can only INPUT electrical signals. They can read digital or analog signals.

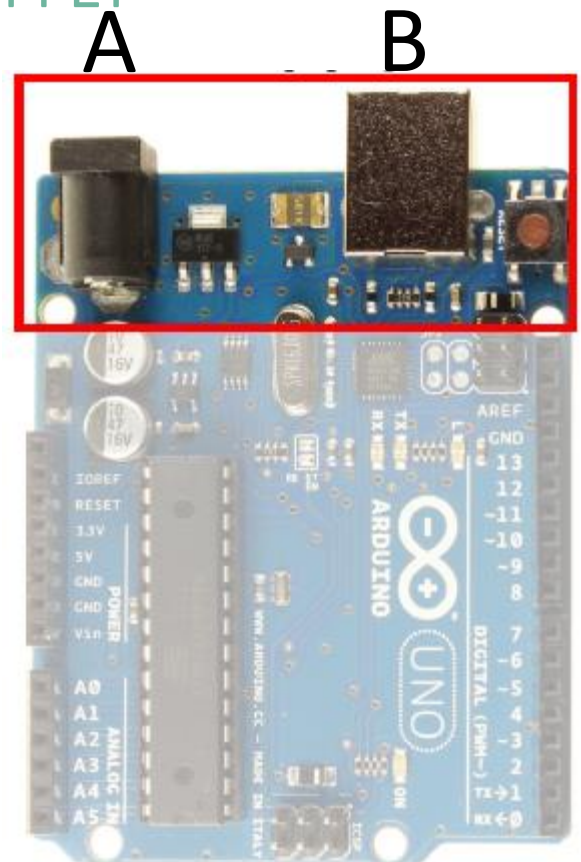
- An Analog Signal, unlike a digital signal, can take on more values than mere on or off, so we can read values that vary between all the way off and all the way on.
- These pins are useful for sensors that can measure many values, such as a light sensor, a distance sensor or a potentiometer (or knob).
- Note that these pins are only used for INPUT (reading information) and not for OUTPUT (writing information).



(4) POWER SUPPLY

This is where we can connect a battery pack, power cable or USB Cable to provide power to the Arduino.

- Port A is where you connect a battery pack. The Arduino has a built in regulator that runs on 7V to 12V.
- The Vin pin matches the voltage supplied to the board.
- Be careful not to give too much power to devices connected to Vin because you may burn out the devices.
- Port B is the connection for the USB cable. Not only can the USB cable supply power to the Arduino, but it is also how we send the code from the computer to the Arduino.



What is a breadboard and why do we use them?

This section explains the breadboard. Most Arduino activities use a breadboard. Breadboard lets a smooth connection between plugging wires and electronic parts. The link is not as sturdy as a permanent connection, such as soldering, but it makes it easy to move pieces around when something is not working. For instance, when building a project for the first time, no permanent connections is needed until knowing that the circuit works.

Breadboards also make it easier to connect multiple wires, which is especially helpful when using multiple devices that need power. All breadboards do this by plugging wires into small holes attached to a metal strip inside the breadboard. The strip connects either rows or columns on the breadboard, and any parts plugged into holes that touch the same metal strip will be electronically connected. (To see how the breadboard makes connections between the holes, take off the back of a breadboard and look inside).

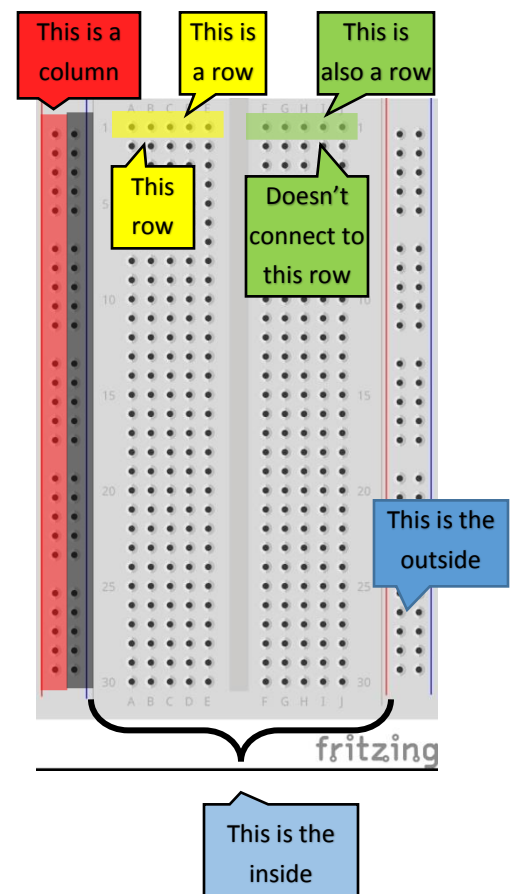
There are two different types of breadboards: (1) Standard breadboard, and (2) mini breadboards. Our Arduino activities use a standard breadboard.

Standard Breadboards

A standard breadboard is one that has both rows and columns of connections. The "inside" of the board connects the rows and the "outside" of the board connects the columns. A single row is defined as 5 holes adjacent to each other and NOT across the gap in the middle.

A standard breadboard usually come with letters and numbers along the top and sides to help with wire placement. Each number connects to itself, but does not join across the middle aisle. So, 1a connects to 1b, 1c, 1d, and 1e (the yellow highlighted row in the image) but does not connect to 1f, 1g, 1h, 1i, or 1j (the green highlighted row in the image). Likewise, 1a does not connect to 2a or 3a or any other a.

Standard breadboards also come with a column (or rail) on each side, with a long strip of positive (highlighted in red in the image) and negative (highlighted in black in the image) holes. Although there are gaps along the line, the entire column is connected. That way, both positive and negative rails can be used to connect many devices that need power.



Mini Breadboards

Since many of the activities use minimal components, it is possible to complete them using a mini breadboard. Notice that a mini breadboard is a subset of the standard breadboard. They do not include the outside rails and have about half as many rows. When using a mini breadboard, it may be easier to color a few rows in different colors (as shown here) to assist with proper connections.

