

Getting Started with Arduino

5 Minute Start

1. Download the Arduino IDE [1.5k](#) version 1.0.5.
 2. Connect Arduino module to computer via USB cable
 3. Snap power module + battery and cable to any of the 3 inputs on the Arduino module [USB does not power module].
 4. Snap together any needed input, output or wire modules.
 5. Upload sketch!
-

Step by Step Tutorial

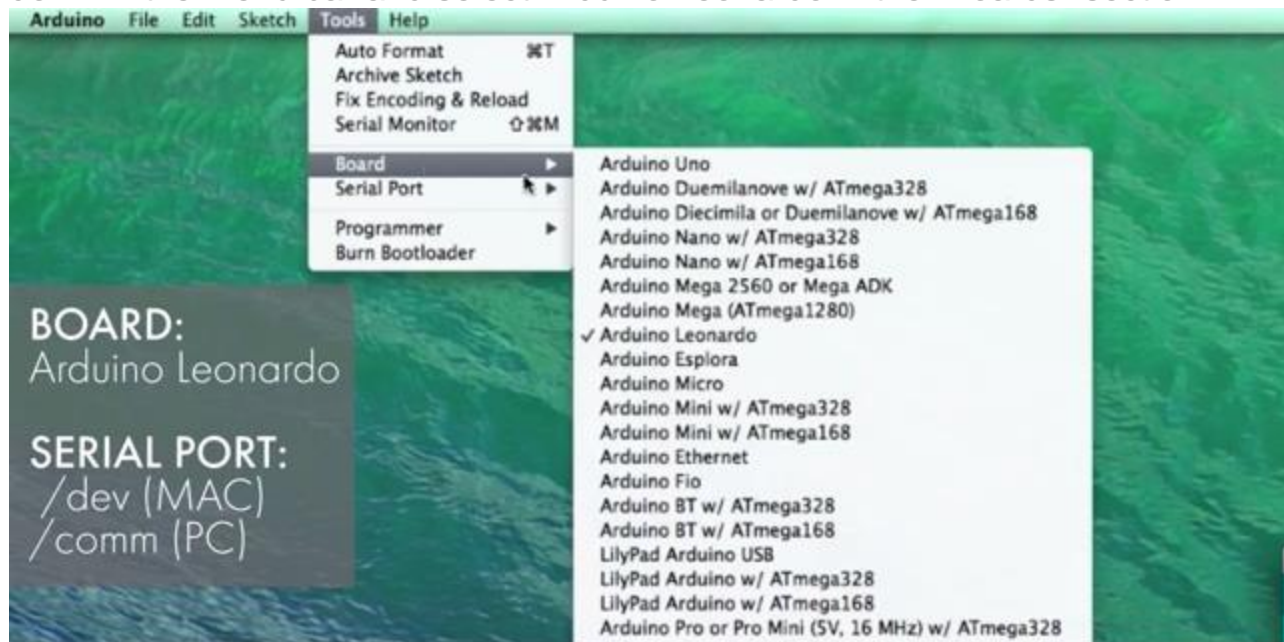
The first thing you will need to do is download the Arduino IDE [1.5k](#) (Integrated Development Environment). This is the software program used to write code ("sketches" in Arduino terms) and upload it to the Arduino module. The Arduino IDE is free to download and can be found here [1.5k](#), both for Mac and PC.

Once you've installed the Arduino IDE, you can open up a sketch and start coding. If you are new to programming, we recommend you go through some of the sketch project tutorials [888](#) we have posted. Start with the Blink sketch [658](#) for example.

Once you have the sketch window open, you can connect your Arduino module to the computer via a micro USB cable. Unlike conventional Arduino boards, the USB connection only provides a data connection. Like all littleBits modules, the Arduino module requires a power module to function. This includes when you are uploading a sketch.

With the Arduino module powered, the computer should now recognize a connection and you might see the RX LED flash. You'll now need to select what kind of Arduino you are using in the menu bar. Go to the "Tools" drop

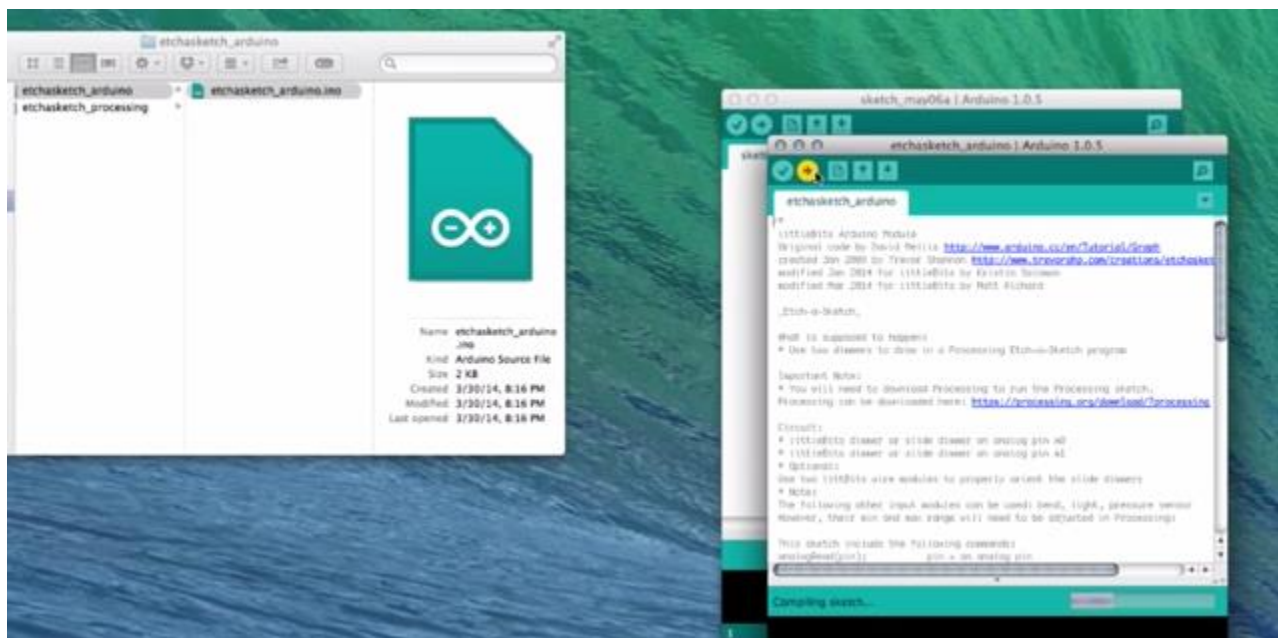
down in the menu bar and select Arduino Leonardo in the "Boards" section.



Now select the serial port the Arduino will communicate over by going back to "Tools" and selecting the correct port in the "Port" section. The name of the port will depend if you are on a Mac or Windows computer. On a Mac, it will start with /dev/tty.usbmodem... and a PC will start with COMM... In Windows, you can look for the USB serial device in the ports section of the Windows Device Manager. If you are on a Linux machine, the port will look like /dev/ttyUSB...



Once these are selected, you simply need to press the "Upload" button in the toolbar at the top of the sketch window.



Detailed information about navigating the Arduino IDE can be found here.

When you start uploading, you will see a message that says "Compiling sketch...", then the message will read "Uploading...". The TX and RX LEDs

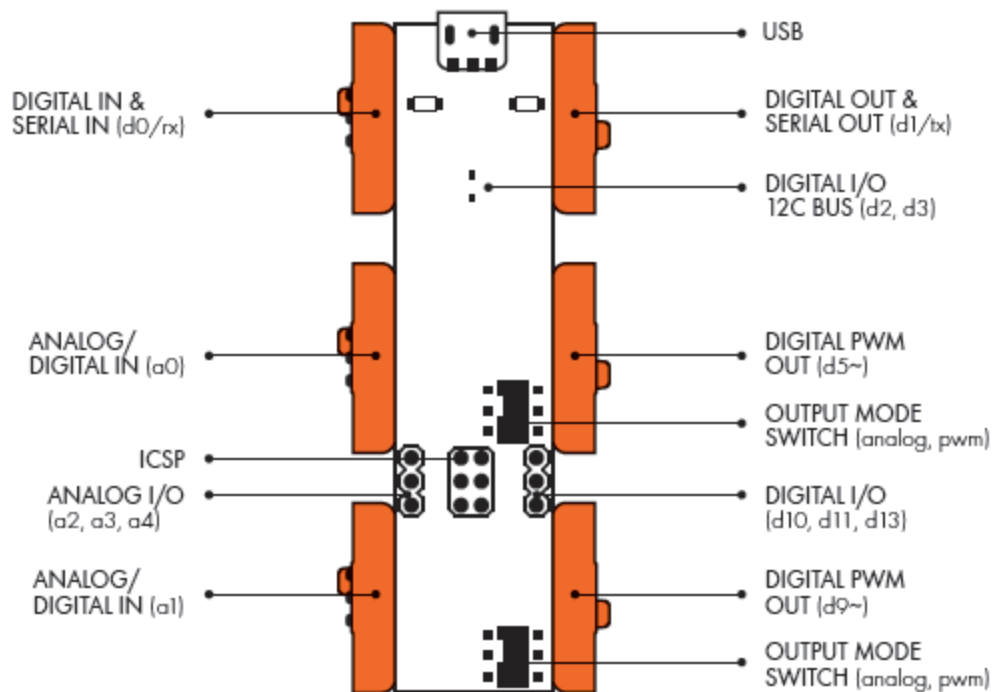
should flash and the message will read “Done uploading.” Uploading the sketch shouldn’t take more than 10 seconds.

Occasionally the uploading process gets hung up. Click the Upload button again and it will restart the upload process. On some occasions, there will be an error. You can read the debug message to solve the issue.

Powering your littleBits Arduino Module

Unlike conventional Arduino boards, the USB connection only provides a data connection so a littleBits power module is still required. Like any littleBits circuit, you can place an input module after the power to alter the signal before it gets to the Arduino module. Using multiple inputs at one time means you'll need a fork or branch to supply power to the other input modules that comes before the Arduino module.

littleBits Arduino module Pins and Features



There are six microcontroller pins connected to our bitSnap™ connectors. On one side of the board, there are three input bitSnaps™. They are referred to as D0, A0, and A1 in the Arduino environment. D0 is a digital pin that is also a Serial input (known as RX). A0 and A1 are analog inputs that can also be used as digital inputs.

The other side of the board features three output bitSnaps™. D1 is a digital pin that is also a Serial out (known as TX). D5~ and D9~ are PWM outputs that are also capable of simulating an analog DC voltage (more on that later).

Both the TX and RX pins have LEDs associated with them so show the status of Serial communication between the board and the computer.

For advanced Arduino users, we added a few features that can be taken advantage of and offer connections not previously offered on a littleBits module. They are:

I2C - The top side of the PCB has two pads which break out pins D2 and D3 from the ATmega32U4. These are the SDA and SDL lines used in I2C communications so multiple boards can be chained together. There are unpopulated pads for 10K pull up resistors if I2C implementation is needed. These pins can also be used as GPIO.

ICSP - There is a standard ICSP connection in the form of through holes that can be soldered to with jumper wires or male/female headers. The bootloader can be updated/changed through these pins and programs can be uploaded via a AVR programmer.

Analog GPIO - Pins A2, A3, and A4 are available in the form of through holes that can be soldered to with jumper wires or male/female headers.

Digital GPIO - Pins D10, D11, and D13 are available in the form of through holes that can be soldered to with jumper wires or male/female headers.