

A slideshow thrown together
from bits and pieces.

Kind of at the last minute too.

Arduino

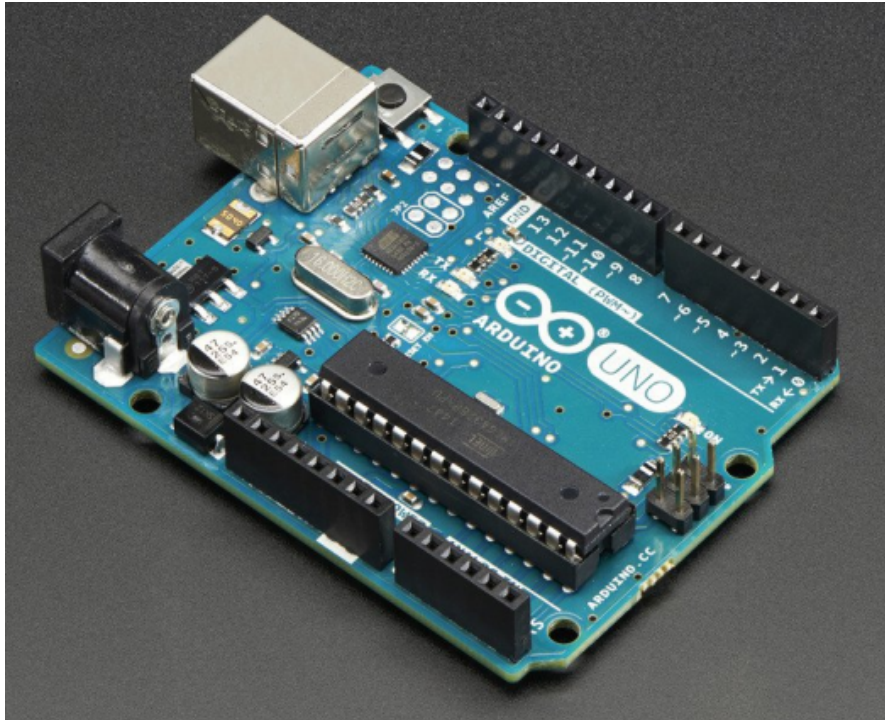


Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects.

It is NOT an android.

It uses its own language which is similar to C+, Python, and Java.

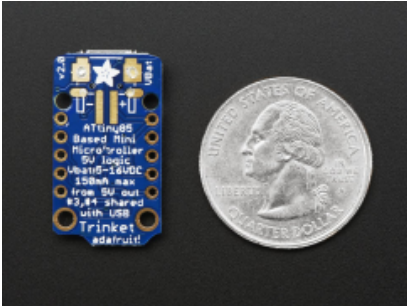
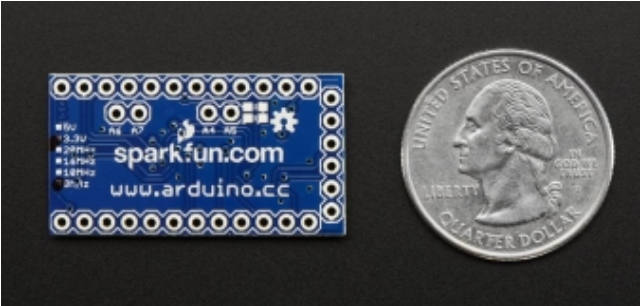
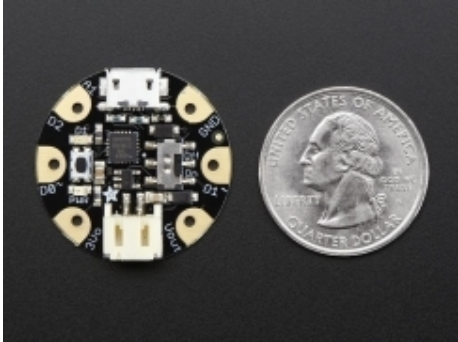
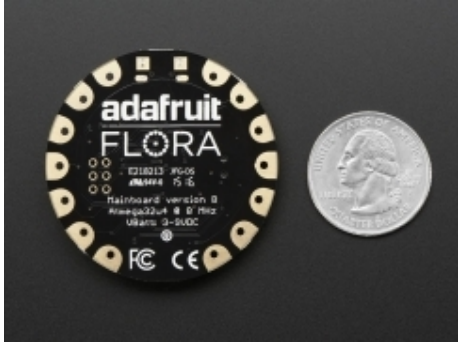
Arduinos come in many shapes and sizes.



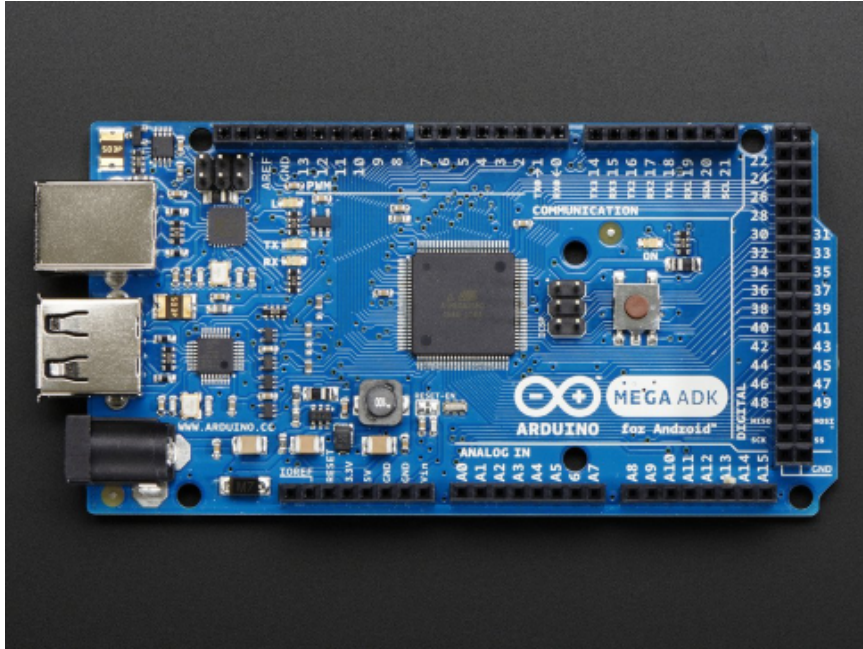
The basic 24 pin board is about the size of a deck of playing cards.

It can be powered via USB or 9v battery.

Here are some examples of smaller ones



And a much larger one!



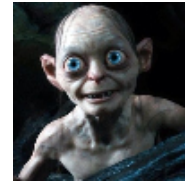
WOW!

54 in/out digital pins!

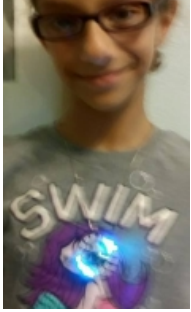
16 analog pins!

256k onboard memory!

We wants it!



Some (not so) practical applications



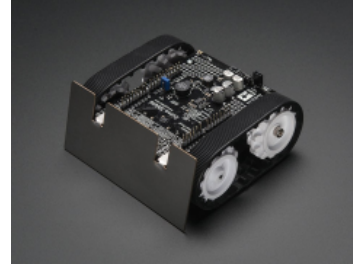
Animated jewelry



Spiderman web shooter



Robots



More robots



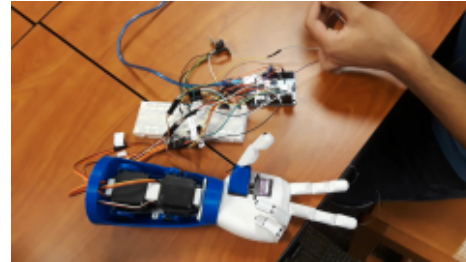
Supercool drum kit



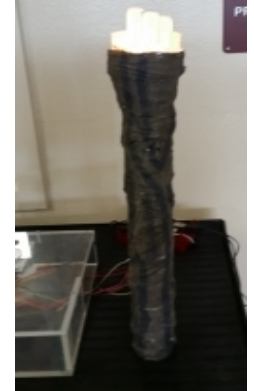
Temp monitor



Lightsabers
'Cause everyone should have a lightsaber.



Even More robots



Torches for Theatre

Inputs

Buttons

Dials

Switches

Mics

Light sensors

Pressure sensors

Programmed script

Outputs

Digital display

Audio

Lights

Motors

Servos

Wifi

Bluetooth

How to get some!

Authorized OCPS vendor:

Adafruit.com

Pros- They have EVERYTHING!

And lessons and sample code and tons more.

Cons- You have to pay a minimum \$9.00 to get something Shipped

On the sly:

Amazon.com

Pros- They have even more stuff than adafruit.

Cons- You may have difficulty purchasing.
Not so much assistance included.

Standards - you know you want some...

[G.K12.5.2.1b](#)

Problem Solving - Understand: Recognize and emulate effective implementation of creative problem solving skills.

[CTE-IT.68.PROG.08.07](#)

Explain strategies used in problem-solving, and relate them to computer programming.

[CTE-TECED.68.ROBTEC.02.07](#)

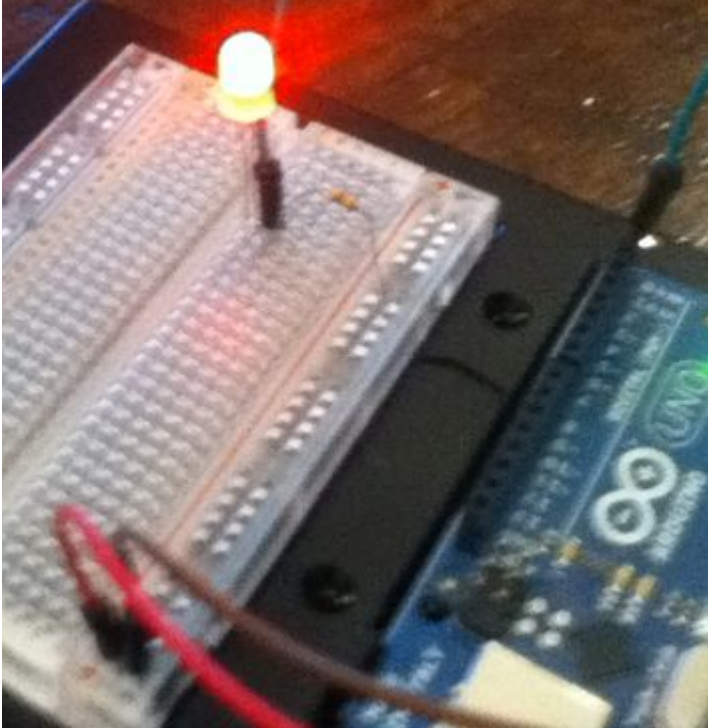
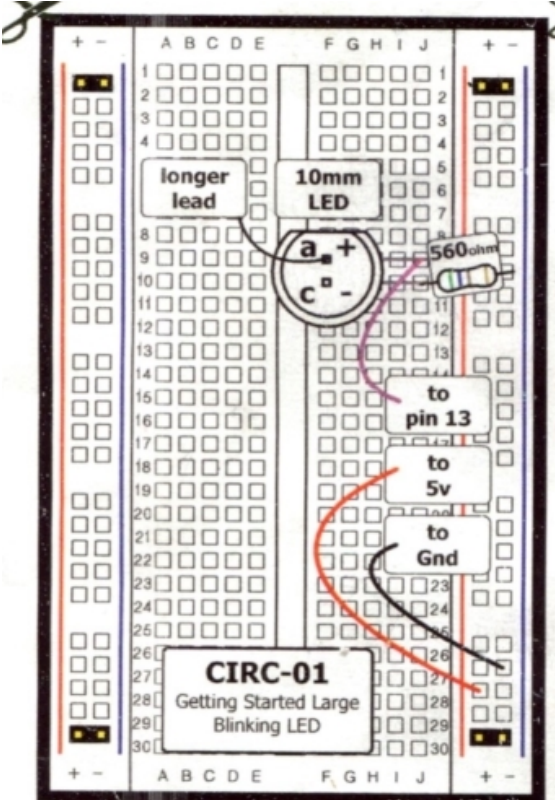
Demonstrate the use of testing and debugging in the problem solving process.

[SC.35.CS-CP.2.3](#)

Create a program using arithmetic operators, conditionals, and repetition in programs.

And more!

Enough chit chat- let's make something!



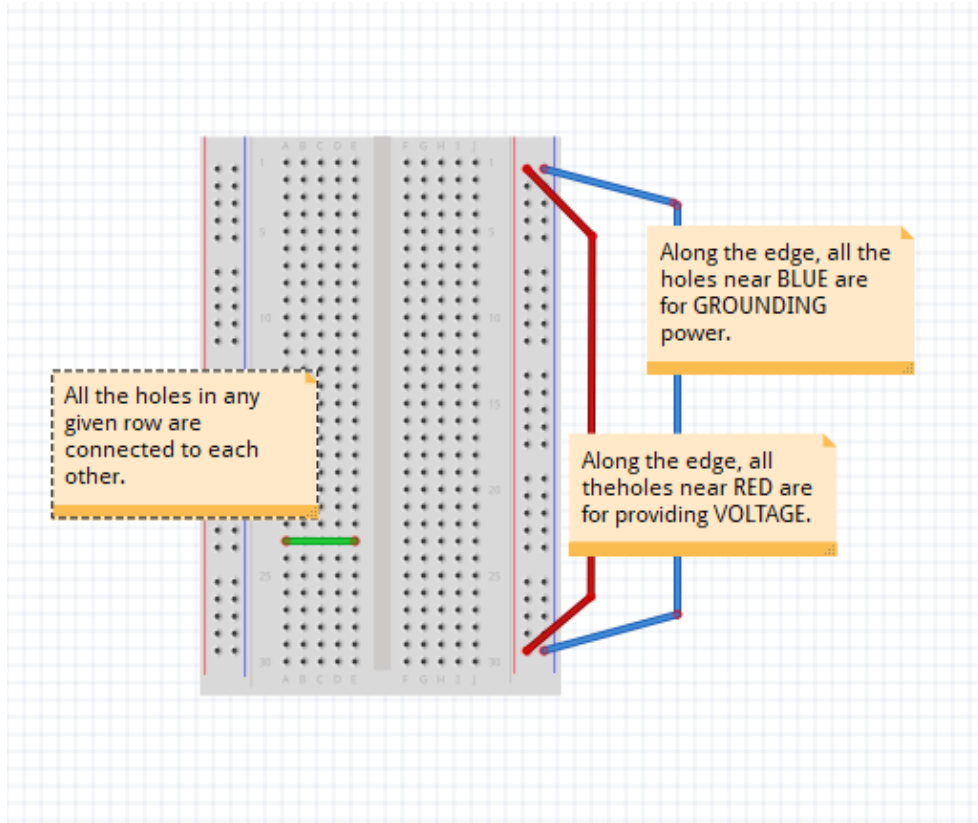
Order of operations (cheat sheet)

1. Plug in your physical Arduino to a PC via USB.
2. Launch the Arduino software.
3. On the top menu, click Tools.
4. From the drop down, check that your BOARD is the type you are using.
5. From the drop down, click PORT and choose the active COM port.

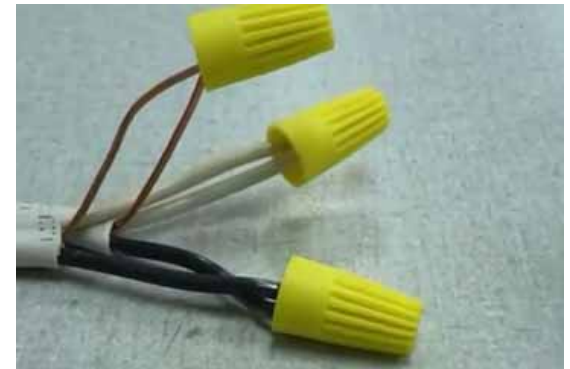
After your project is ready

6. Are your wires set up properly? (Power to power, ground to ground, resistors)
7. Does your pin setup in your code match your pin setup in reality? (pin 4 is declared, is pin 4 on the board what you are plugged into?)
8. Click upload.

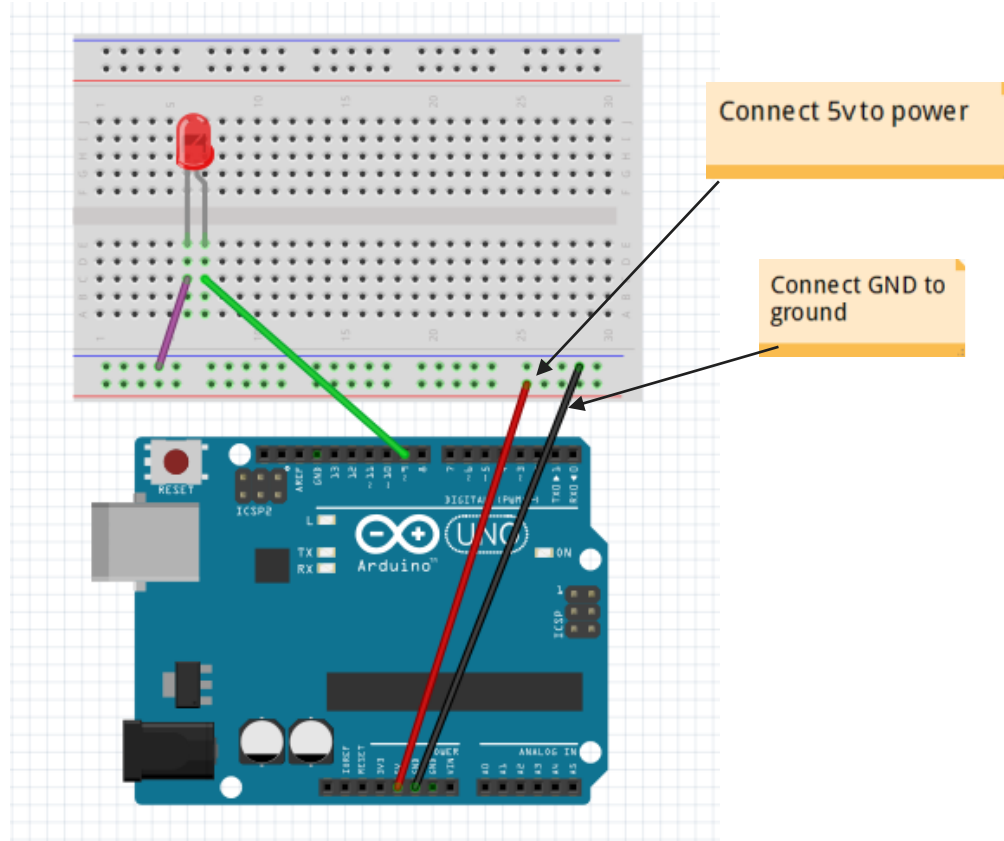
Your Breadboard



Saves you from having to do this!



Your 1st build.



The color of your wires doesn't matter, just stay organized!

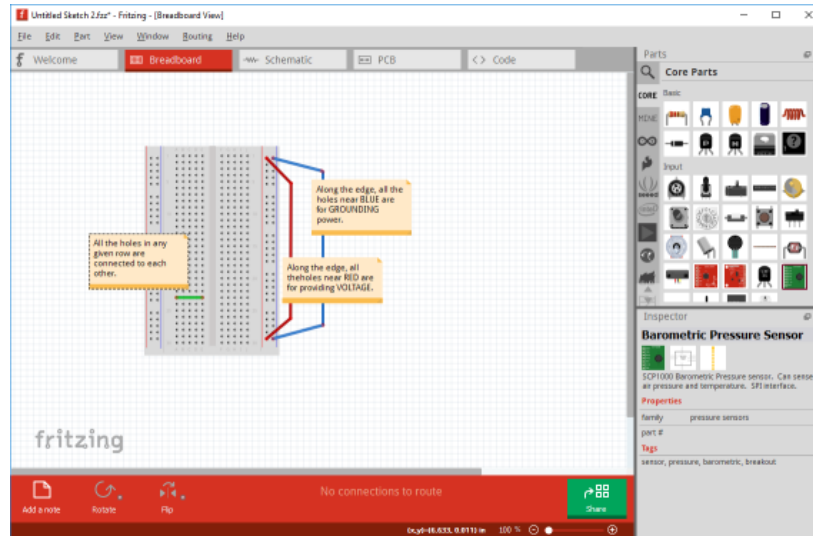
A side note....

How did I make those awesome diagrams?


I use Fritzing!

It's free software for setting up electronics. It has libraries with drag and drop images of all sorts of things you may use in a makerspace electronics lab.

Fritzing.org



Now let's program it!

1. On the Arduino software, click File, Examples, 01 Basics, Blink.
2. Any code behind a “/*” or a “//” are comments for folks like yourselves, they do not affect the program. After I read them, I like to erase them to keep my code clean.
3. Code will be in Teal, Orange, or Black.
4. All lines need to end with a ;
5. All segments need to be between { and }
6. Click the  to upload the code. (It won't work, but that's on purpose!)

Now HACK it!!

Can you:

Make it blink faster?

In a pattern?

Add another light that blinks on it's own?

Now for something completely different:

<http://www.oomlout.com/a/products/ardx/>

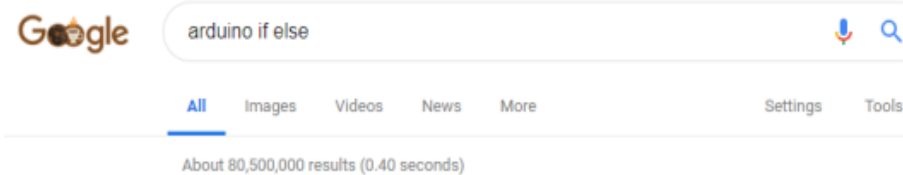
This site has plans and diagrams and sample code for a variety of projects.

May I recommend: circ-03, circ-04, circ-06, circ-07, circ-08, circ-09?

Need more help?

Arduino.cc has literally everything arduino related.

I've found that googling “arduino *****” will find me the help I need with coding.



Else - Arduino

<https://www.arduino.cc/reference/en/language/structure/control-structure/else/>

An **else** clause (if at all exists) will be executed if the condition in the **if** statement results in false
When a true test is found, its associated block of code is run, and the program then skips to the line following the entire **if/else** construction.



A screenshot of the Arduino.cc website's documentation page for the 'else' statement. The page has a teal header with navigation links: HOME, STORE, SOFTWARE, EDU, RESOURCES, COMMUNITY, HELP. A search icon is in the top right. The main content area has a left sidebar with a navigation menu: LANGUAGE, FUNCTIONS, VARIABLES, STRUCTURES, LIBRARIES, GLOSSARY. The main text area contains a note: "Note that an **else if** block may be used with or without a terminating **else** block and vice versa. An unlimited number of such **else if** branches is allowed." Below this is the "Syntax" section with a code block:

```
if (condition)
{
  // do thing A
}
else if (condition)
{
  // do thing B
}
else
{
  // do thing C
}
```

 This is followed by an "Example Code" section with a code block:

```
if (temperature >= 70)
{
  //Danger! Shut down the system
}
else if (temperature >= 60 && temperature < 70)
{
  //Warning! User attention required
}
else
{
  //Safe! Continue usual tasks ...
}
```

 At the bottom, there is a "See also" section with a link to "if statement".