



# CIRCUITS AND CODE: FUN WITH ARDUINO!

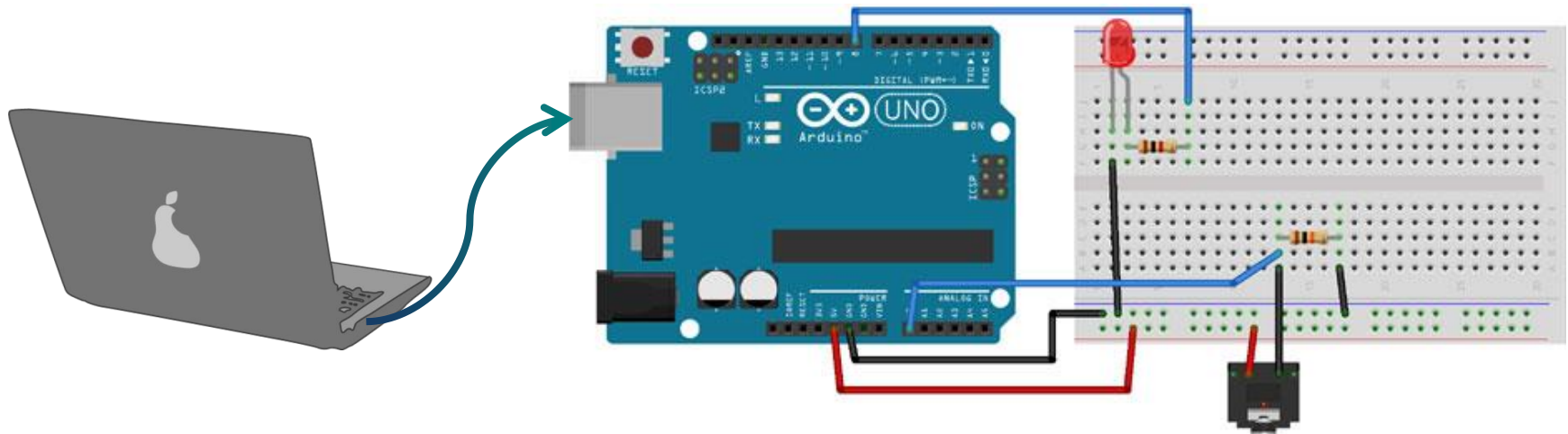
NATALIA BAKLITSKAYA

[NATALIA.BAKLITSKAYA@GMAIL.COM](mailto:NATALIA.BAKLITSKAYA@GMAIL.COM)

# What is Arduino?

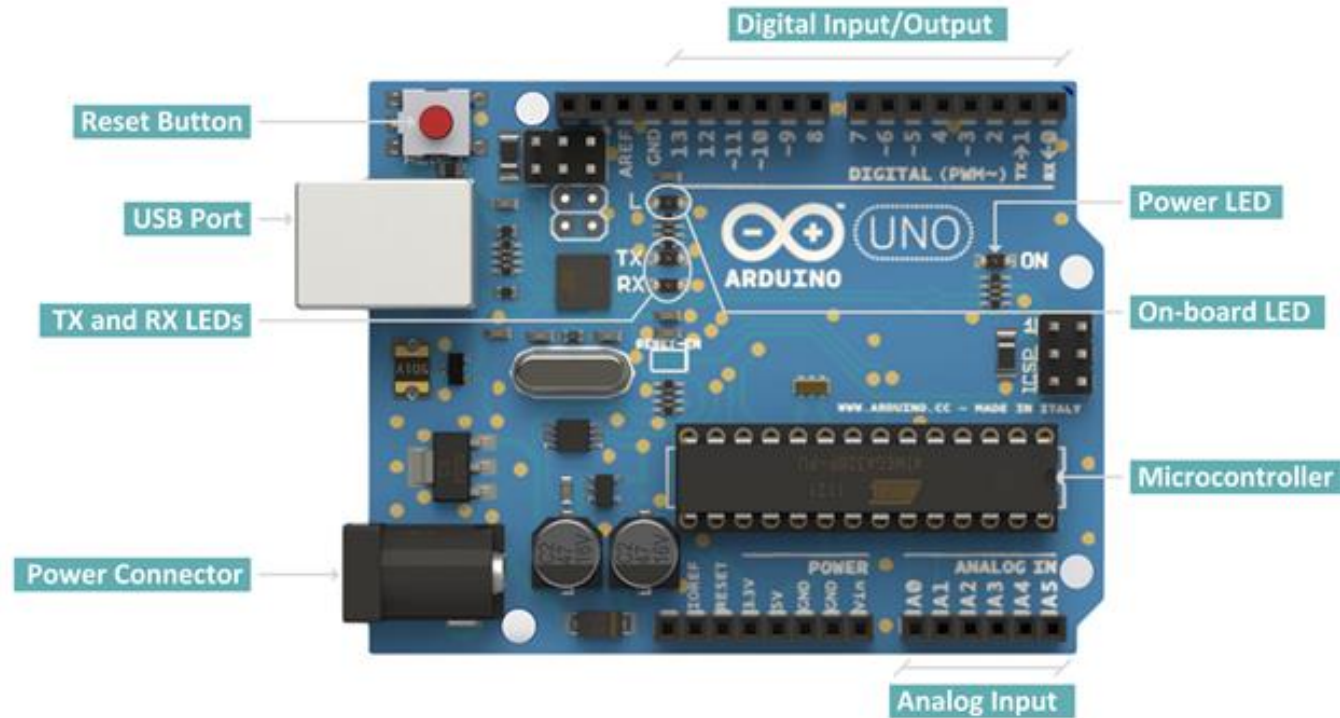
The heart (or the brain) of DIY projects

- Microcontroller
  - A board with a chip that can be programmed to do many different things
- Inputs (digital and analog)
  - Read input from sensors like ultrasonic sensor and thermistor
- Outputs
  - Drive Circuitry



# What is Arduino?

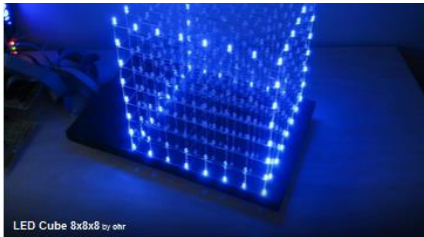
The heart (or the brain) of DIY projects



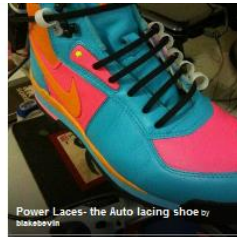
# What Can I Build With Arduino?

- <http://www.instructables.com/id/20-Unbelievable-Arduino-Projects/>

Here are twenty amazing Arduino projects that you almost wouldn't believe, if not for that they are the real deal. These authors have turned their wildest dreams into reality with the power of Arduino, an easy-to-use microcontroller development board. It is no wonder that Arduino literally translates to "Strong friend (masculine)" in Italian. Anything is possible with the mighty power of Arduino. It's compact, it's straightforward, and makes embedding electronics into the world-at-large fun and easy. Check out some of these amazing projects, and get inspired to build your own reality.



LED Cube 8x8x8 by chr



Power Laces- the Auto facing shoe by blakebevin



Plantduino Greenhouse by oloversredure



The EyeWriter 2.0 by thesystemz



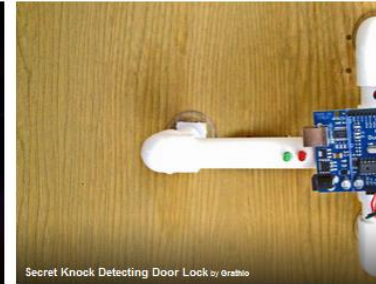
Twitter Mood Light - The World's Mood in a Box by RandomMatrix



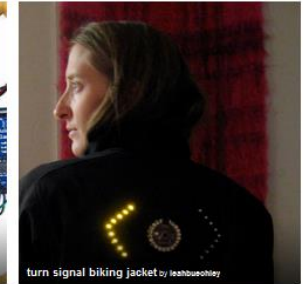
Flamethrowing Jack-O'-Lantern by randof



Make a 24X6 LED matrix by systemx



Secret Knock Detecting Door Lock by oratio



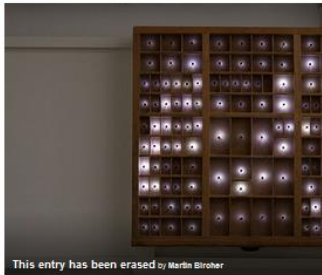
turn signal biking jacket by leahbushley



Tree Climbing Robot by ben\_k



Rave Rover - Mobile Dance Stage by owilliamson



This entry has been erased by Martin Birchler



Sigh Collector by mkontopo

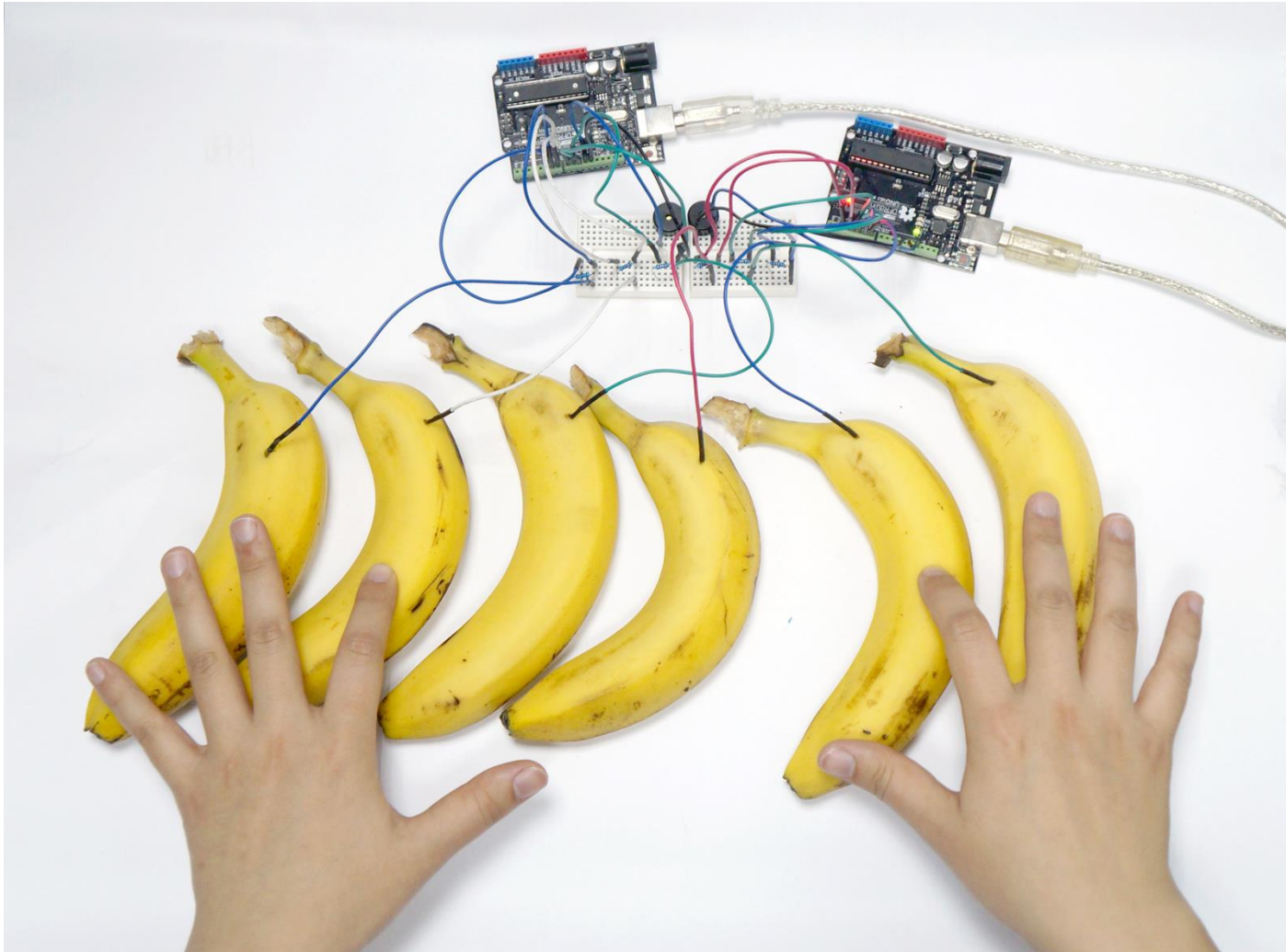


Make a Fire Breathing Animetronic Pony from FurReal Butte... by iv\_Lee



---

# What Can I Build With Arduino?

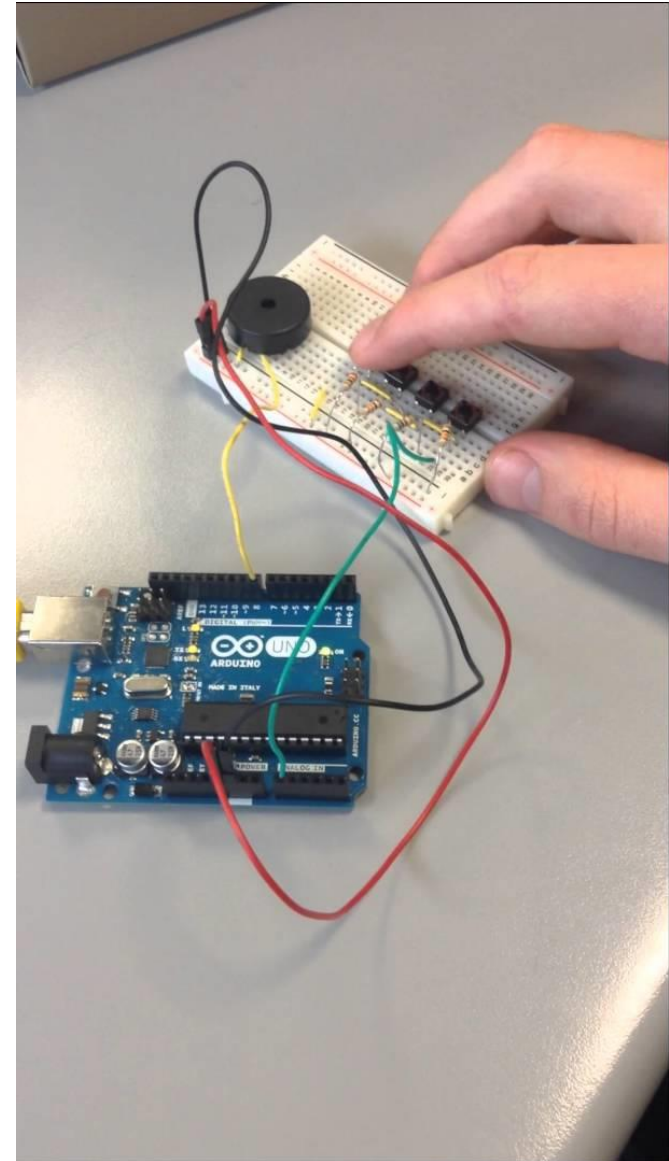


# What are you working with?



# What Can I Build With Arduino?

- **Spaceship Interface**
- **Love-O-Meter**
- Color Mixing Lamp
- Mood Cue
- Light Theremin
- Keyboard Instrument
- Digital Hourglass
- Motorized Pinwheel
- Zoetrope
- Crystal Ball
- Knock Lock
- Touchy-feely Lamp



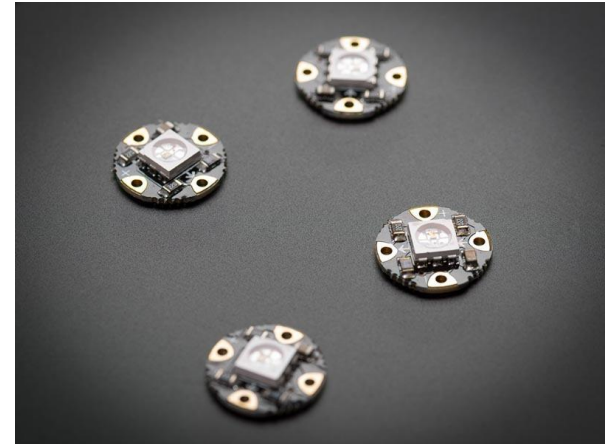
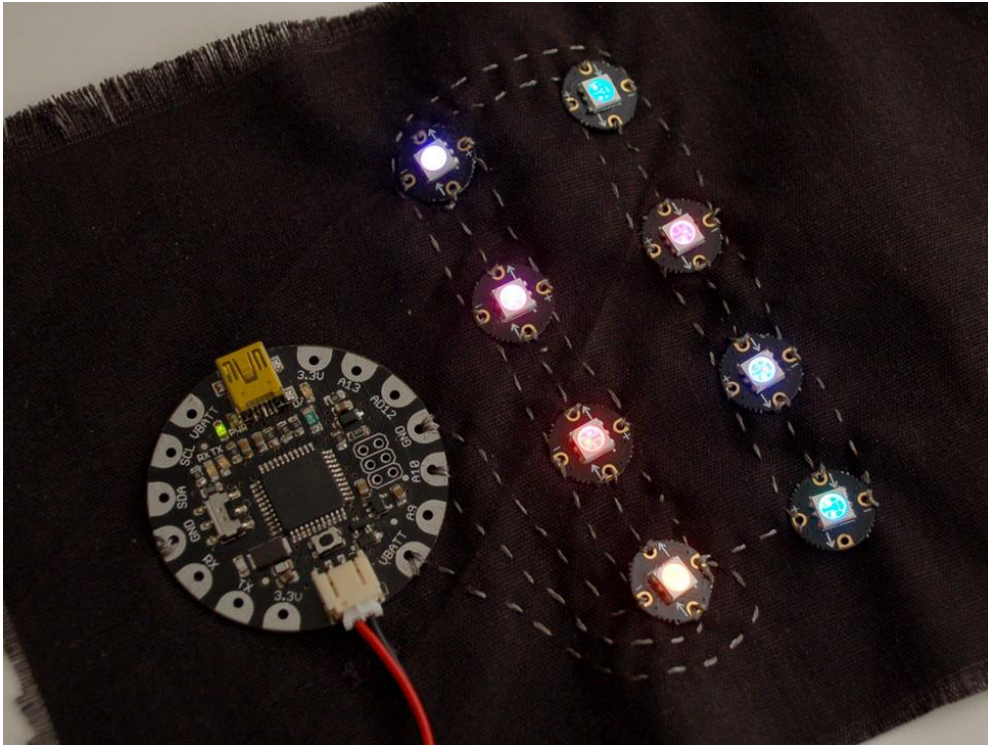
---

# Who are you working with?

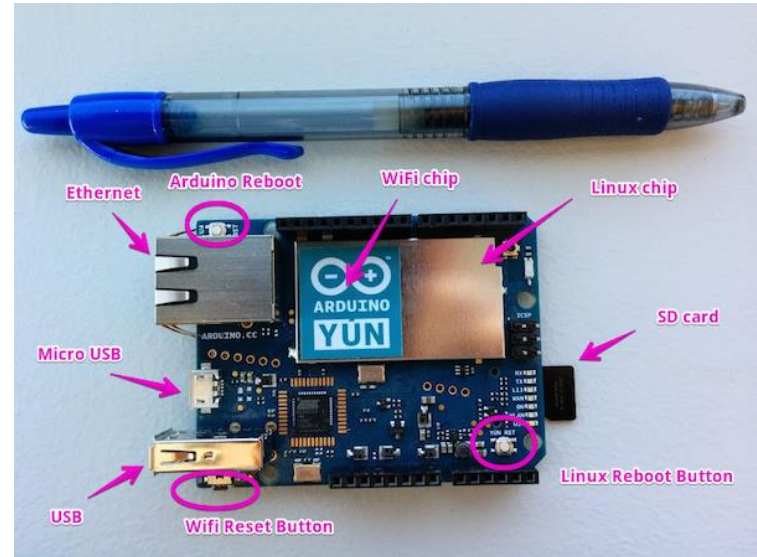
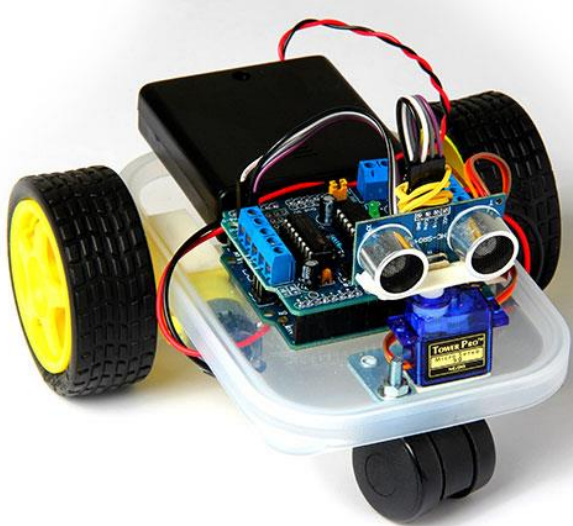
- Groups
- Teaching Assistants!



# NeoPixels for Makers!



# But what about Robots and IoT?



“..Once you’ve finished this tutorial, you may want to put your newly acquired skills to use by:

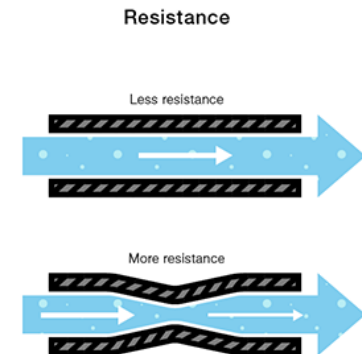
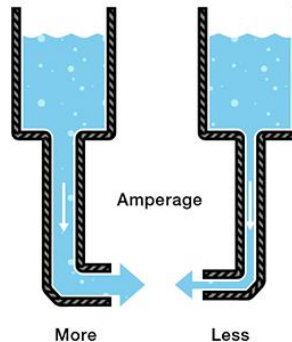
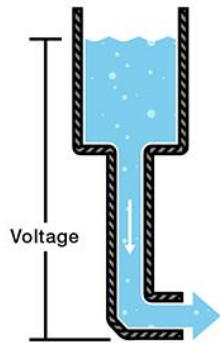
[Building an Arduino Yun powered photo booth](#)

[Sending SMS and MMS from your Arduino Yun](#)

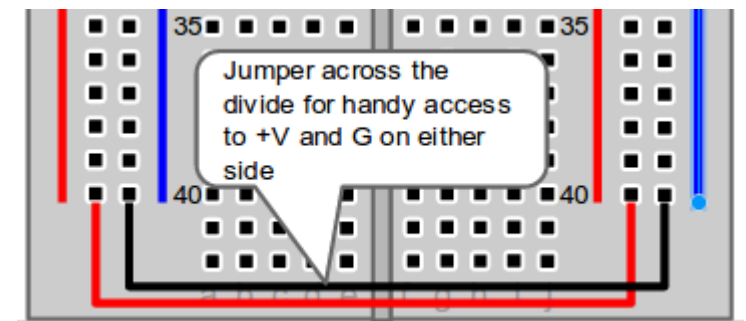
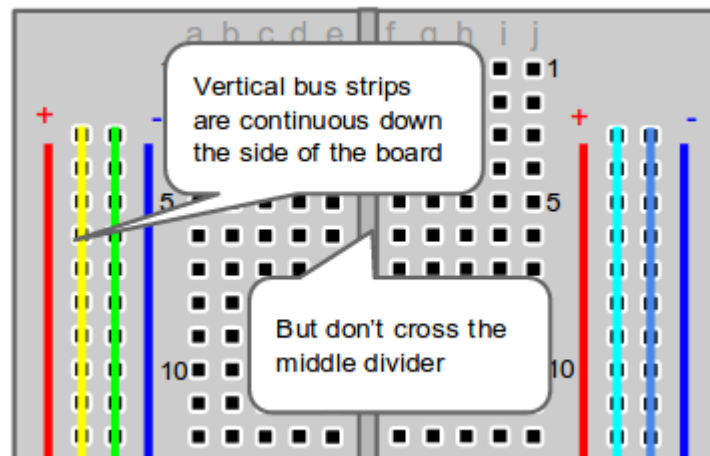
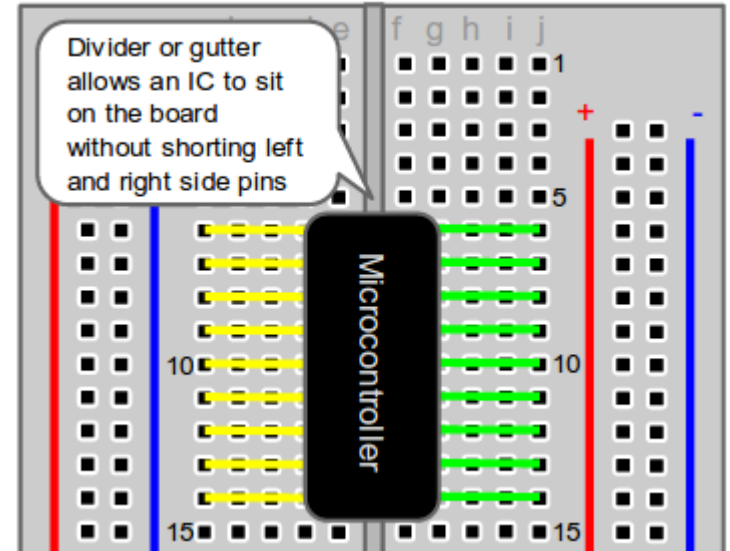
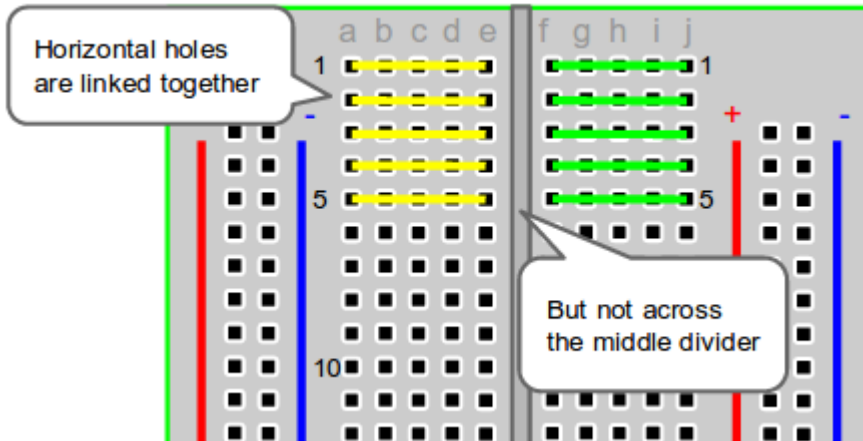
[Teaching your dog to send you selfies with the Arduino Yun](#) “

# Voltage, Current, Resistance

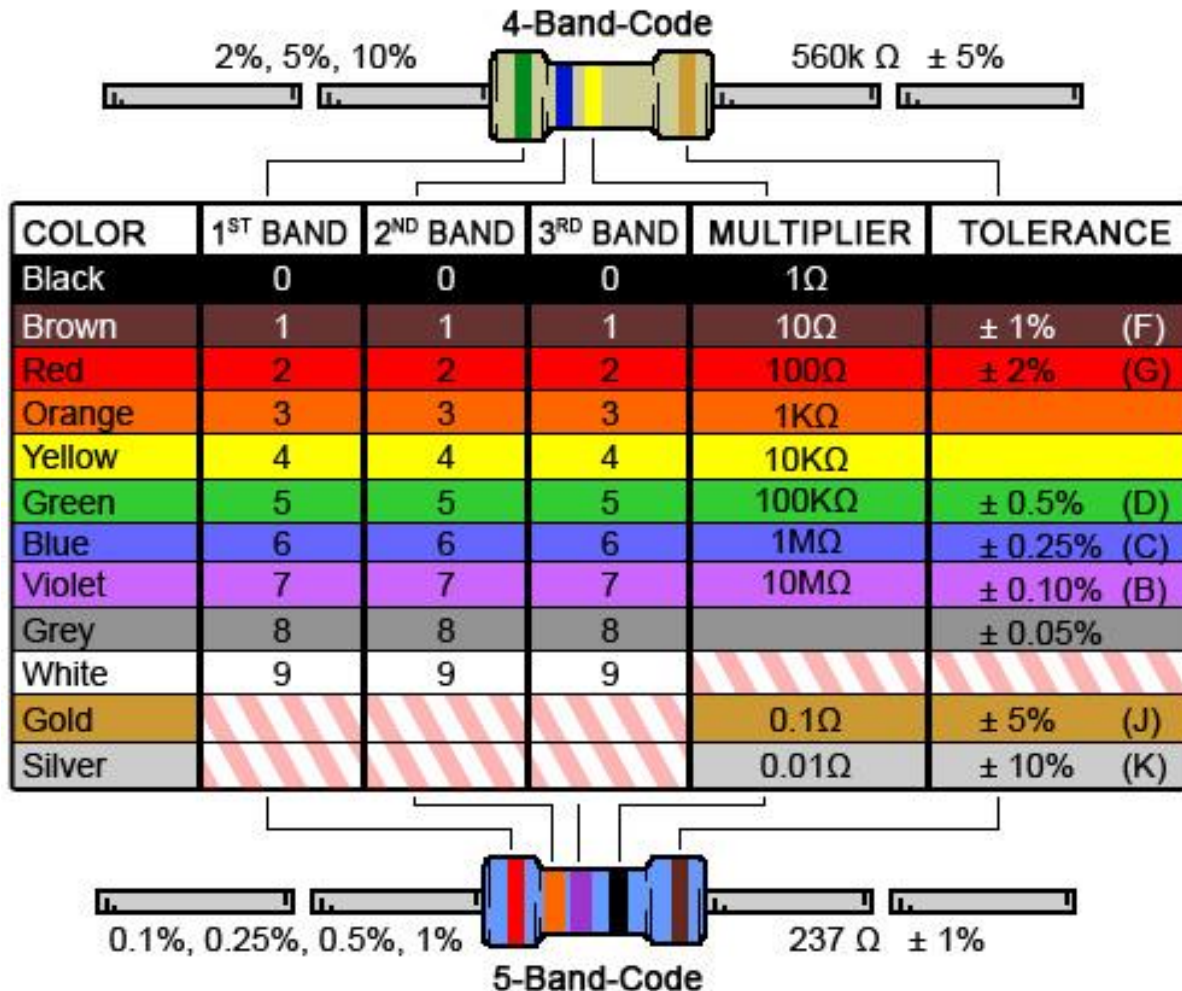
- **Voltage** is the difference in charge between two points.
- **Current** is the rate at which charge is flowing.
- **Resistance** is a material's tendency to resist the flow of charge (current).
- $V = I * R$



# Breadboard Pins



# Resistor Color Codes





---

# Digital Logic Levels + Analog Pins

- OFF = 0 = 0 Volts
- ON = 1 = 5 Volts
- Analog pins have 1024 levels between 0 Volts and 5 Volts

# Arduino Code: setup() and loop() Functions

## ARDUINO APPLICATION LIFECYCLE



### **Runs only for one time**

- We initialize system here

### **Runs forever in a loop**

- Like while(true)
- All of our logic resides here



# Functions

Segmenting code into functions allows a programmer to create modular pieces of code that **perform a defined task** and then **return to the area of code from which the function was "called"**.

The typical case for creating a function is when one needs to perform the same action multiple times in a program.

## Anatomy of a C function

Datatype of data returned,  
any C datatype.

"void" if nothing is returned.

Function name

Parameters passed to  
function, any C datatype.

```
int myMultiplyFunction(int x, int y){
```

```
int result;
```

```
result = x * y;  
return result;
```

Return statement,  
datatype matches  
declaration.

Curly braces required.

```
}
```

---

# Functions

Standardizing code fragments into functions has several advantages:

- Functions help the programmer stay organized. Often this helps to conceptualize the program.
- Functions codify one action in one place so that the function only has to be thought out and debugged once.
- This also reduces chances for errors in modification, if the code needs to be changed.
- Functions make the whole sketch smaller and more compact because sections of code are reused many times.
- They make it easier to reuse code in other programs by making it more modular, and as a nice side effect, using functions also often makes the code more readable.
- There are two required functions in an Arduino sketch, `setup()` and `loop()`. Other functions must be created outside the brackets of those two functions. As an example, we will create a simple function to multiply two numbers.

Try this example located in a folder on your Desktop: **p02\_SpaceShipInterfaceFunctions**