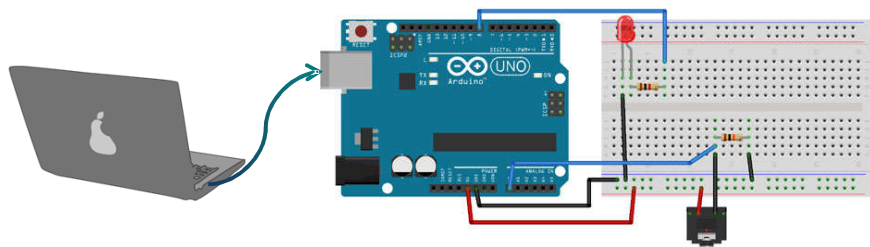




What is Arduino?

The heart (or the brain) of DIY projects

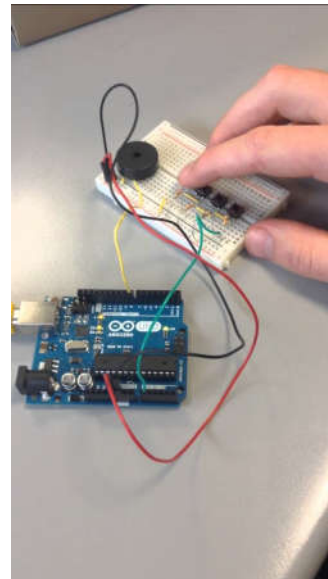
- Microcontroller
 - A board with a chip that can be programmed to do many different things
- Inputs (digital and analog)
 - Read input from sensors like ultrasonic sensor and thermistor
- Outputs
 - Drive Circuitry



Made with  Fritzing.org



- **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

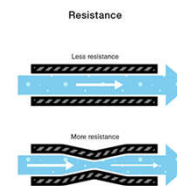
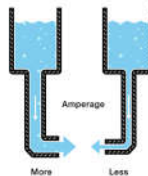
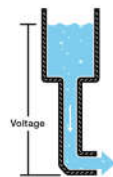


Let's Dive In!

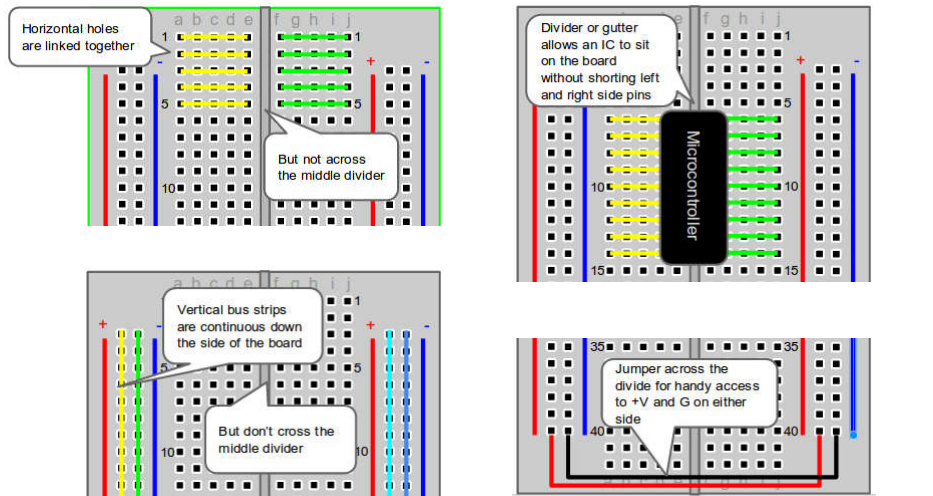
- Open up the Arduino Starter Kit and work through projects 1-3
- Show a TA your completed work for each project

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).
- $V = I * R$

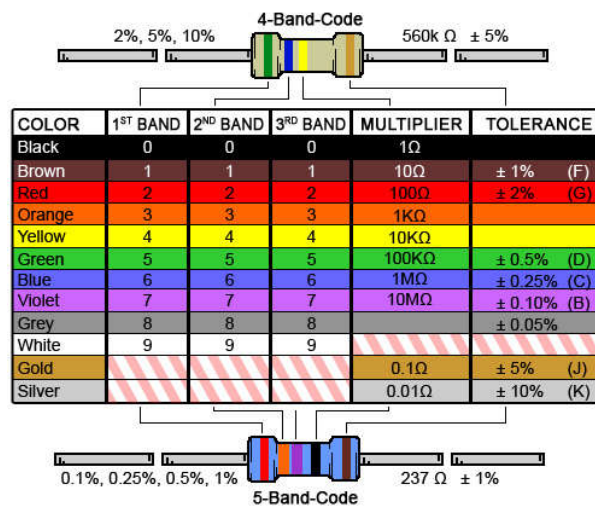


Breadboard Pins



<http://computers.tutsplus.com>

Resistor Color Codes

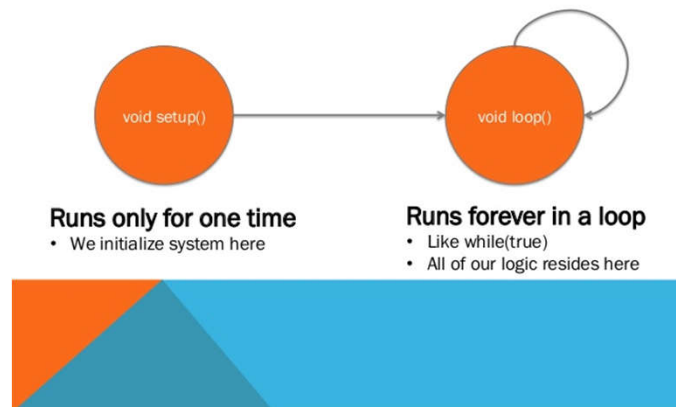


Digital Logic Levels + Analog Pins

- 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



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

```

int myMultiplyFunction(int x, int y){
  int result;
  result = x * y;
  return result;
}

```

Datatype of data returned, any C datatype.
 "void" if nothing is returned.

Function name

Parameters passed to function, any C datatype.

Return statement, datatype matches declaration.

Curly braces required.

• <https://www.arduino.cc/en/Reference/FunctionDeclaration>

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**

• <https://www.arduino.cc/en/Reference/FunctionDeclaration>

Adafruit and LadyAda

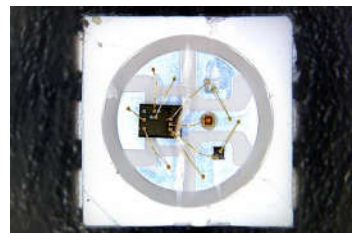
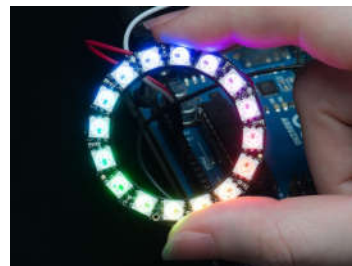
- Adafruit was founded in 2005 by MIT hacker & engineer, Limor "Ladyada" Fried. Her goal was to create the best place online for learning electronics and making the best designed products for makers of all ages and skill levels. Adafruit has grown to over 100+ employees in the heart of NYC with a 50,000+ sq ft. factory. Adafruit has expanded offerings to include tools, equipment and electronics that Limor personally selects, tests and approves before going in to the Adafruit store. Limor was the [first female engineer on the cover of WIRED magazine](#), awarded [Entrepreneur magazine's Entrepreneur of the year](#), and was on [the cover of Make: Vol. 57](#). Ladyada was a founding member of the NYC Industrial Business Advisory Council. Adafruit is ranked #11 in the top 20 USA manufacturing companies and [#1 in New York City](#) by Inc. 5000 "fastest growing private companies". Adafruit is featured in Google's [Economic Impact Report](#). Limor was named a [WHITEHOUSE CHAMPION OF CHANGE](#) in 2016. Adafruit is a 100% woman owned company.



NeoPixels

More than your average LED

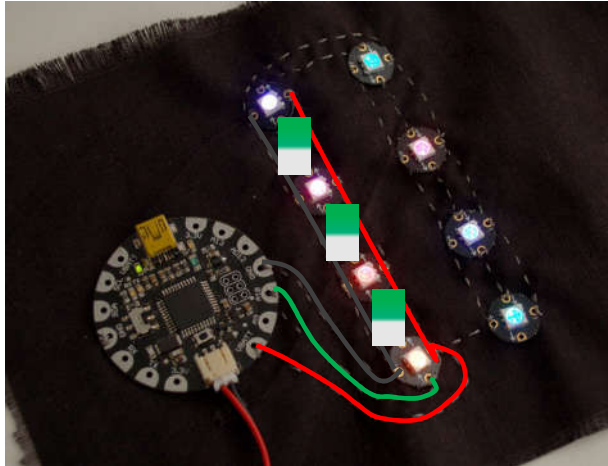
- WS2812
- Dedicated LED driver chip
- Individually addressable
- Use single-wire control protocol



Flora + NeoPixels

For your sweet rave parties. Or practical applications.

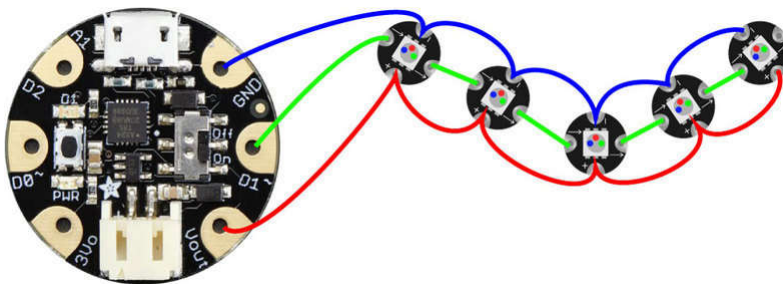
- Ground
- Power
- Din
- Dout



Gemma + NeoPixels

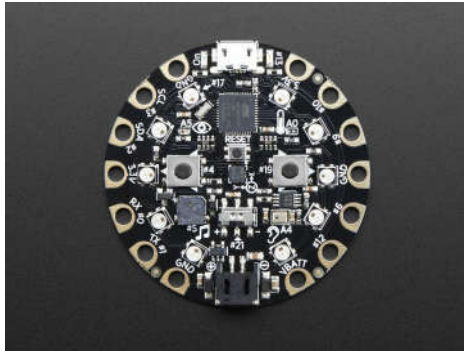
A closer look

- GND, D1, Vout (5V)



Circuit Playground + Alligator Clips + NeoPixel

Modding simple.ino -> GetSET Wearables Packet



GetSET Wearables Packet

Controlling NeoPixels

Use the Adafruit NeoPixel Libraries!

All NeoPixel sketches begin by including the header file:

```
#include <Adafruit_NeoPixel.h>
```

The block of code that follows is mostly descriptive comments. Only the last line is really doing any work:

```
#define PIN 6

// Parameter 1 = number of pixels in strip
// Parameter 2 = pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
// NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
// NEO_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
// NEO_GRB Pixels are wired for GRB bitstream (most NeoPixel products)
// NEO_RGB Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
Adafruit_NeoPixel strip = Adafruit_NeoPixel(60, PIN, NEO_GRB + NEO_KHZ800);
```

Controlling the NeoPixels

Then, in the setup() function, call begin() to prepare the data pin for NeoPixel output:

```
void setup() {
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'
}
```

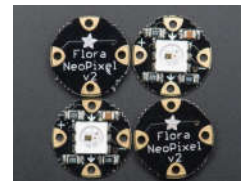
To set the color of a pixel:

```
strip.setPixelColor(n, red, green, blue);
strip.show();
delay(500)
```

Wiring + Code

Which pin is Data?

- Adjust code accordingly to which digital you use to drive the NeoPixels
- Recommended starter program is **simple.ino**
 - Found under File > Examples > Neopixels
 - Adjust PIN (data pin), number of NeoPixels



```
#include <Adafruit_NeoPixel.h>
#include <avr/power.h>

#define PIN 0
#define NUMPIXELS 3

// Parameter 1 = number of pixels in strip
// Parameter 2 = Arduino pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)

Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
```

GetSET Wearables Packet

General code structure

```
#include <Adafruit_NeoPixel.h>
#include <avr/power.h>

// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1
#define PIN          1

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS    2

// When we setup the NeoPixel library, we tell it how many pixels, and which pin to use to send signals.
// Note that for older NeoPixel strips you might need to change the third parameter--see the strandtest
// example for more information on possible values.
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

int delayval = 500 ; // delay for half a second

void setup() {
  pixels.begin(); // This initializes the NeoPixel library.
}

void loop() {

  // For a set of NeoPixels the first NeoPixel is 0, second is 1, all the way up to the count of pixels minus one.

  for(int i=0;i<NUMPIXELS;i++){

    // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
    pixels.setPixelColor(i, pixels.Color(0,150,0)); // Moderately bright green color.

    pixels.show(); // This sends the updated pixel color to the hardware.

    delay(delayval); // Delay for a period of time (in milliseconds).

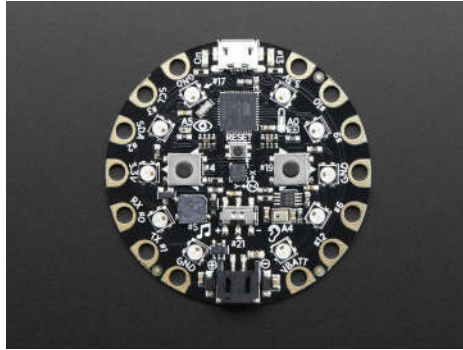
  }
}
```

Strandtest: many NeoPixels, many colors!

- Install IDE, drivers, and NeoPixel library
- Open File > Examples > Adafruit NeoPixels > strandtest
- Modify the code to work with your microcontroller
 - Output pin, number of pixels
- Upload the code to your Circuit Playground microcontroller

Adafruit Circuit Playground

Let's put on our engineering hats!



- Which sensors do we have on board?
- Which sensor are we most interested in trying out?

Adafruit Circuit Playground + Hair Bows

With a little help from Velcro...

Pick up your bow and assemble your wearable!



Fun with Adafruit Circuit Playground

Adjust the code from the GetSET Wearables Packet challenge and upload it to the Circuit Playground

Activate all 10 NeoPixels on board - no alligator clips!

- First, make them light **Red**, **Green**, and **Blue** in *sequence*, with an interval of 200 milliseconds between each NeoPixel.
- Next, make all 10 NeoPixels flash **Red** for 500 milliseconds, followed by **Green**, and then **Blue**.
- Demo your handywork to a TA!
- Finally, adjust **strandtest** code to work on the Circuit Playground!

Fun with Adafruit Circuit Playground

Upload Examples -> Adafruit Circuit Playground -> analog_sensors.ino to the Adafruit Circuit Playground.

- Adjust the code to activate all 10 NeoPixels. Change the MIN and MAX colors for 1 or more values of your choice (R,G,B)
- Try our different sensors and show your TA.

Fun with Adafruit Circuit Playground

Upload some other example programs, like:

1. Birthday_Candles.ino
 2. mega_demo.ino
 3. color_sense.ino
 4. comm_badge.ino
- ...+ customize them!

Another fun demo: <https://learn.adafruit.com/circuit-playground-hourglass/overview>

**Have fun and try more projects
with Circuit Playground!**