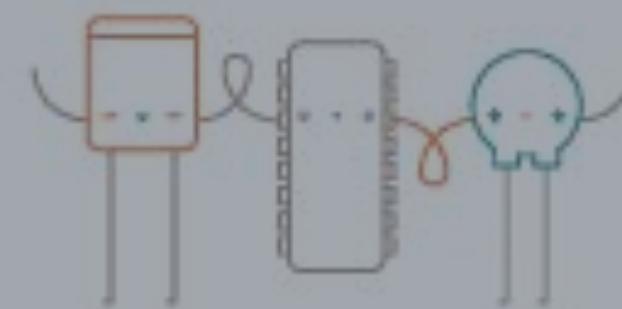




Coding with *Arduino*

AN OPEN PROJECT WRITTEN, DEBUGGED,
AND SUPPORTED BY ARDUINO.CC AND
THE ARDUINO COMMUNITY WORLDWIDE

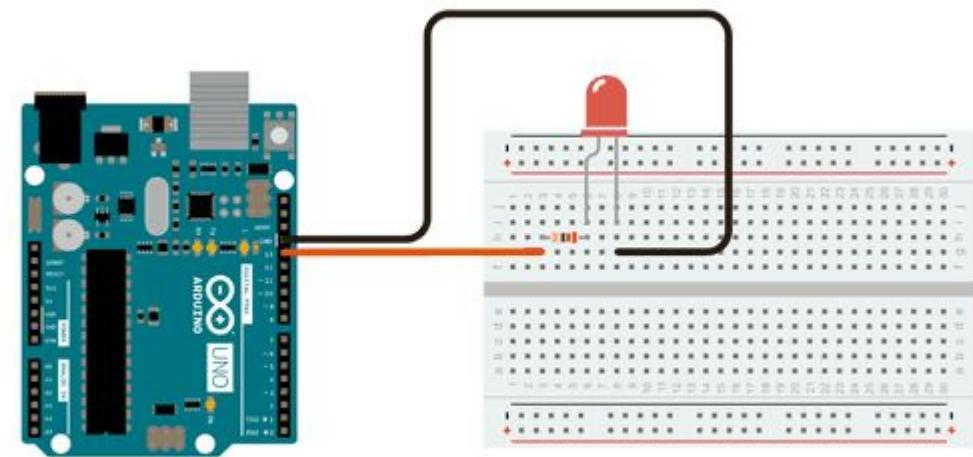
LEARN MORE ABOUT THE CONTRIBUTORS
OF **ARDUINO.CC** on arduino.cc/credits

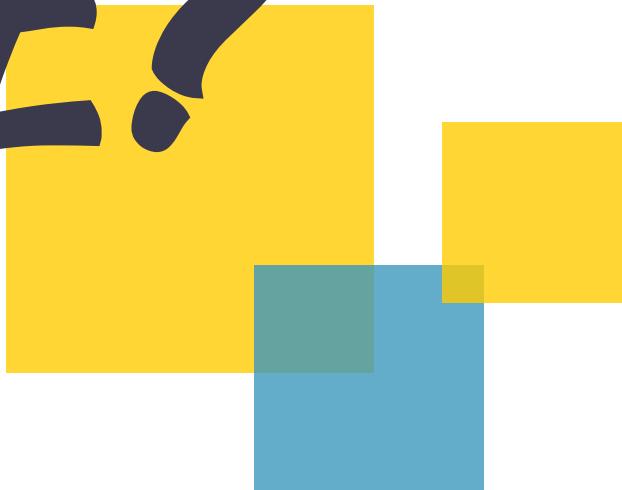


ALLEGHENY COLLEGE COMPUTER AND
INFORMATION SCIENCE DEPT.

Today's Agenda

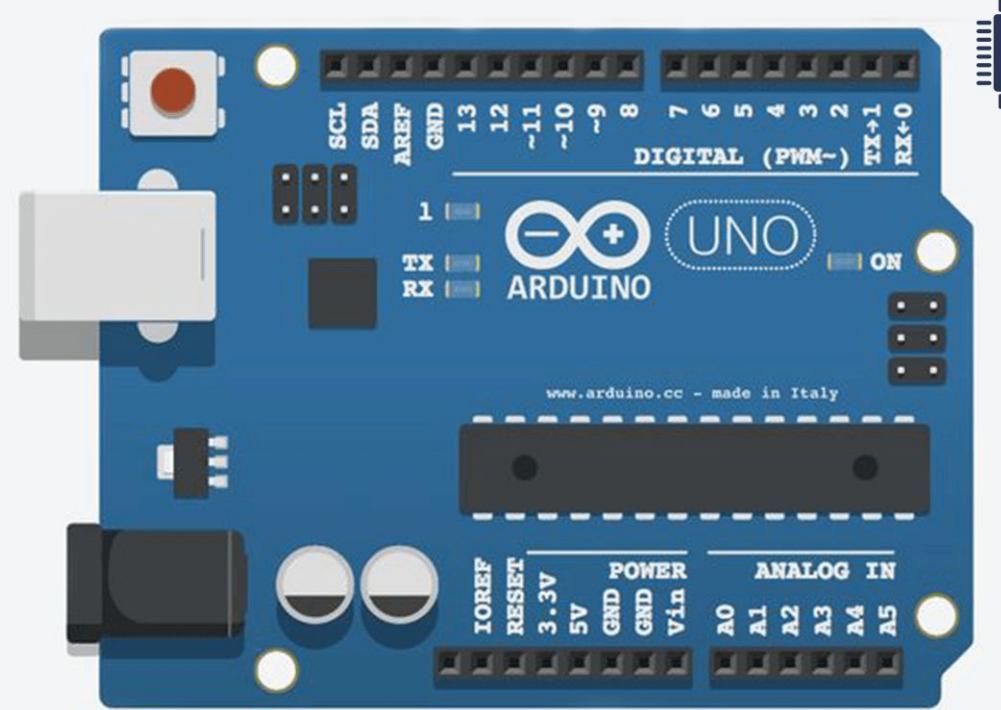
- 01** What is Arduino IDE?
- 02** Arduino Blink
- 03** Arduino RGB
- 04** Arduino Servo





*What is
Arduino IDE?*

- Arduino IDE is how we program the Arduino UNO to perform different operations
- Pros of Arduino IDE:
 - User-Friendly Development Environment
 - Cross-Platform Support: Available for Windows, macOS, and Linux
 - Code Editor and Serial Monitor
 - Built-in Examples and Libraries
 - One-Click Uploading
 - Open-Source and Extensible
 - Support for Multiple Boards



IDE Breakdown

The screenshot shows the Arduino IDE interface. The title bar reads "sketch_aug21a | Arduino IDE 2.3.2". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar has icons for save, upload, and refresh. The board selector dropdown shows "Arduino Uno". The left sidebar has icons for file, folder, book, and search. The main code editor window displays:

```
sketch_aug21a.ino
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8
9 }
10
```

Open up on your computer to follow along

- This is a blank sketch
- Everytime you start a new file it will look like this

IDE Breakdown

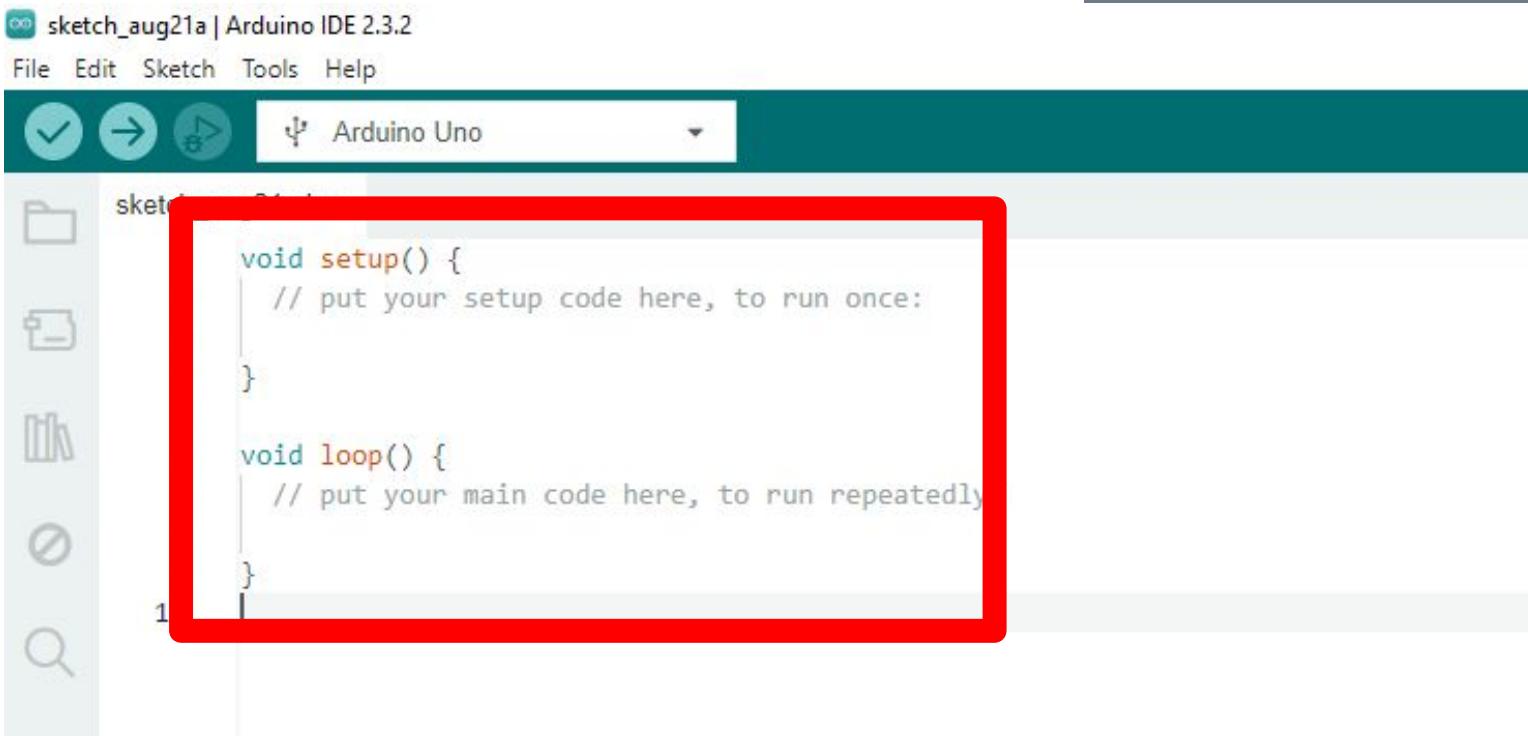
Open up on your computer to follow along



- Everytime you change your code click the checkmark “Verify” button
 - This will compile the code and check it for bugs
 - Tells you if there is an error
- If the verification is successful then click the checkmark to “Upload” the code to the Arduino
 - You will have to reupload after every change
- The panel is where you can select which USB port the Arduino is in
 - If the code does not upload properly it is most likely because the right port is not selected

IDE Breakdown

Open up on your computer to follow along



The screenshot shows the Arduino IDE interface. The title bar reads "sketch_aug21a | Arduino IDE 2.3.2". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for save, upload, and refresh. The board selector dropdown shows "Arduino Uno". The code editor displays the following sketch:

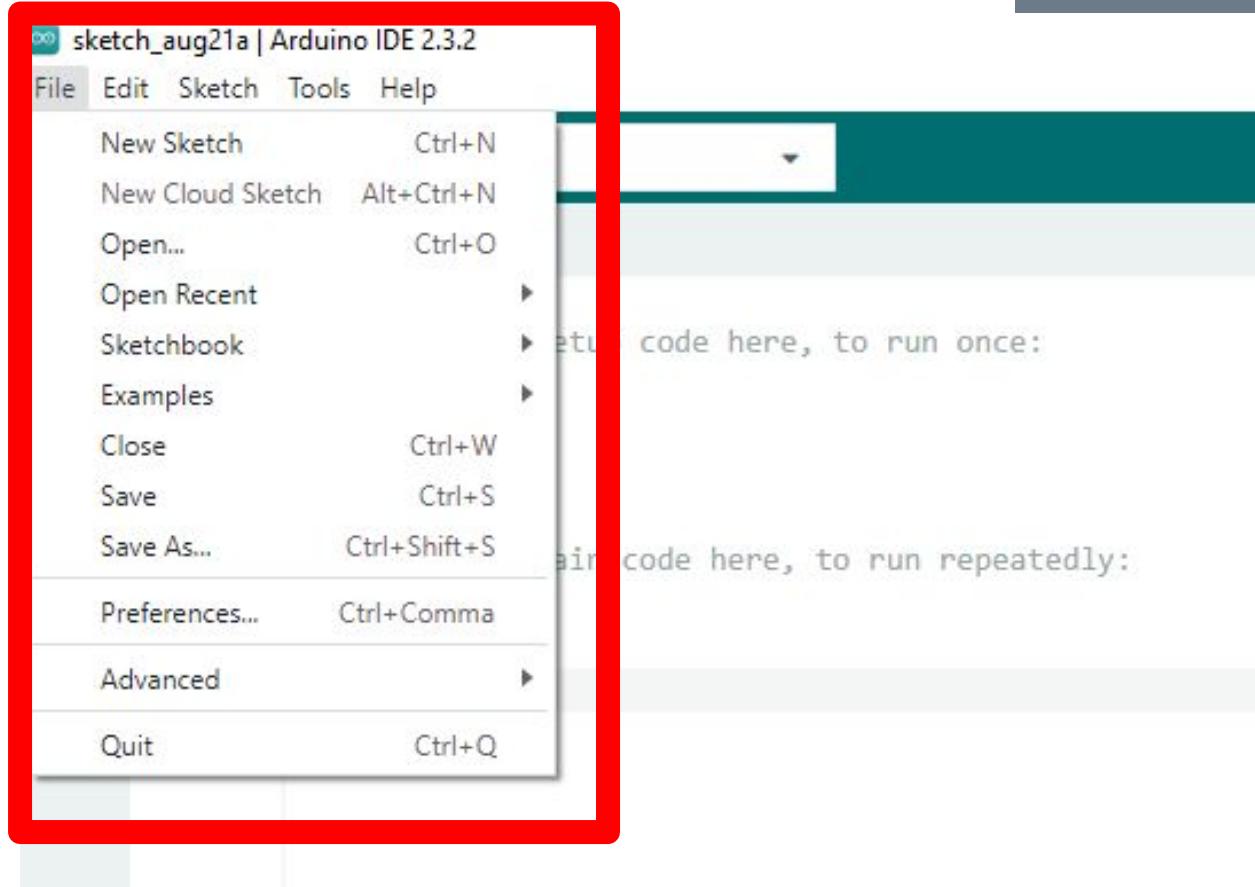
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly
}
```

- The setup and loop functions are the main parts of running your code
- We will go more into detail later but for now know :
 - Setup only runs once and is meant to set up variables and pins used for your program
 - Loop is the main code and where the behavior of the program is defined

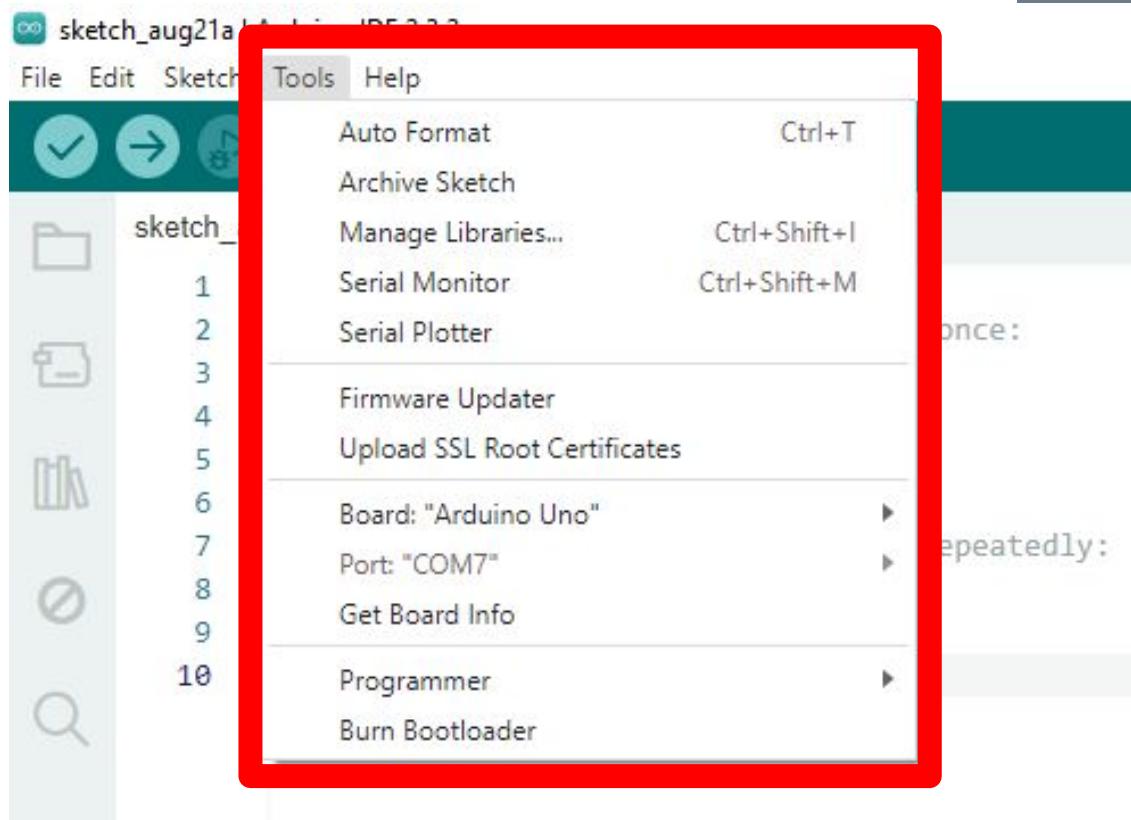
IDE Breakdown



Open up on your computer to follow along

- The file tab works a lot like a Word document
- New Sketch: Open a fresh document
- Open: Open an existing document
- Examples: Open example files from Arduino or other libraries
- Close: Close the current document
- Save: Save your document progress
 - SAVE OFTEN it does not save automatically
- Save as: Save the current document as a new document

IDE Breakdown



Open up on your computer to follow along

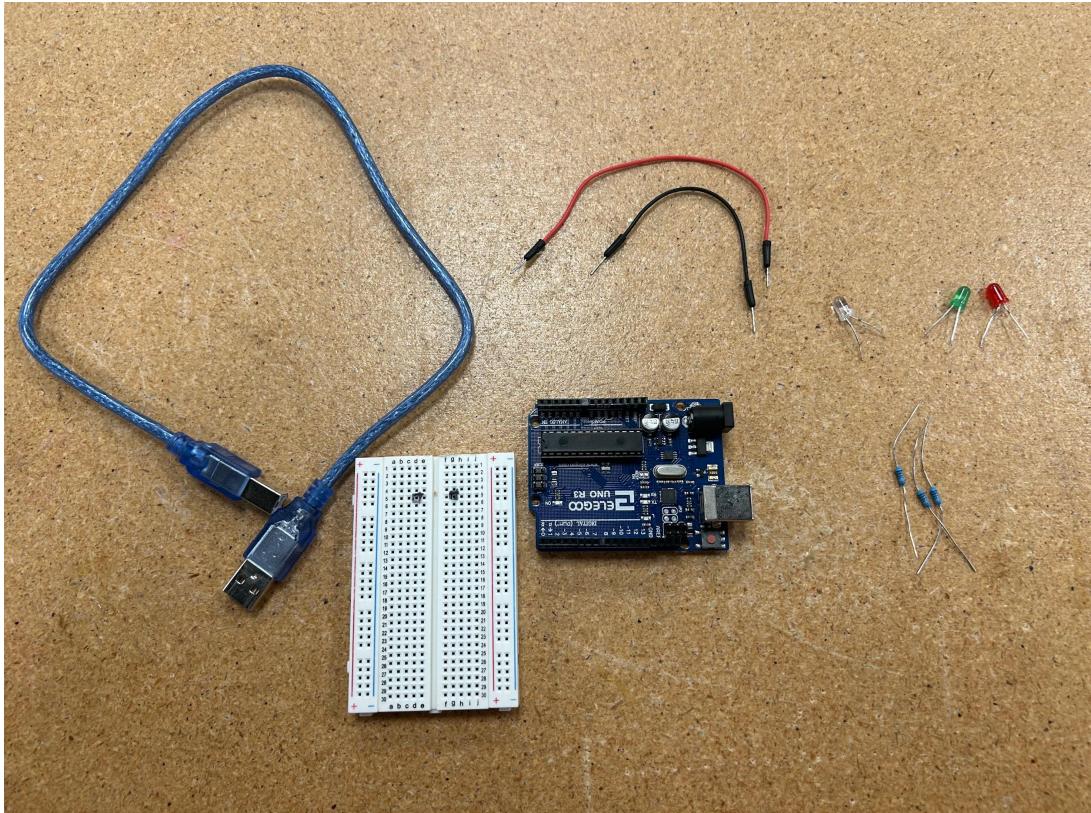
- The tools tab contains a lot of the utility we will need to program
- Manage Libraries: Add code made by other people/companies (Open-source)!
- Serial Monitor: Code can print outputs to the serial monitor to make it visible to the user
 - Good for sensor outputs!
- Serial Plotter: Graphed version of the outputs
- Board: Select the current version of the board you are using here. Arduino Uno is correct.
- Port: This is where you select the USB port your Arduino is in. It can sometimes be automatically selected
 - If code will not upload it is most likely the wrong port is selected!
 - Can also be selected on the main page.



Arduino Blink



Step 1: Gather Materials



- 1 Arduino UNO
- 1 Breadboard
- 3 LEDs of any color
- 3 470 Ω resistor (Yellow - Violet - Brown - Golden)
- Hookup wire
- 1 Arduino USB cord

Breadboards

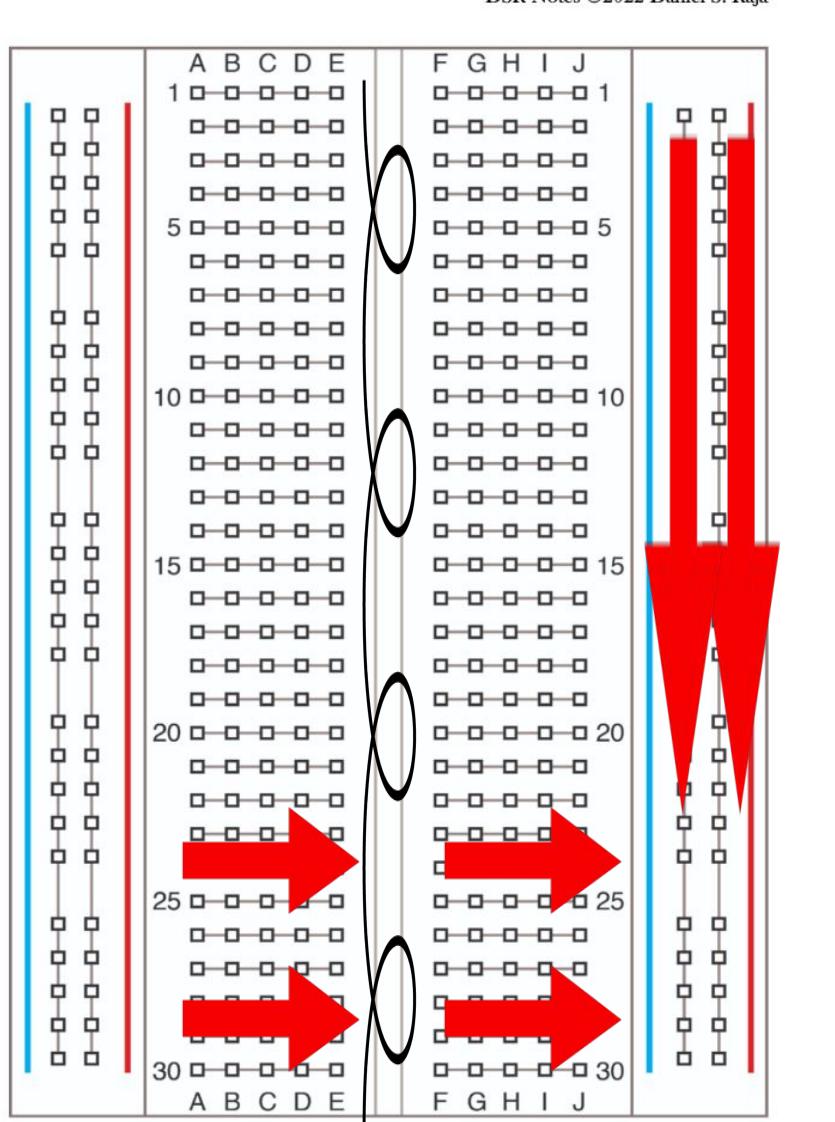
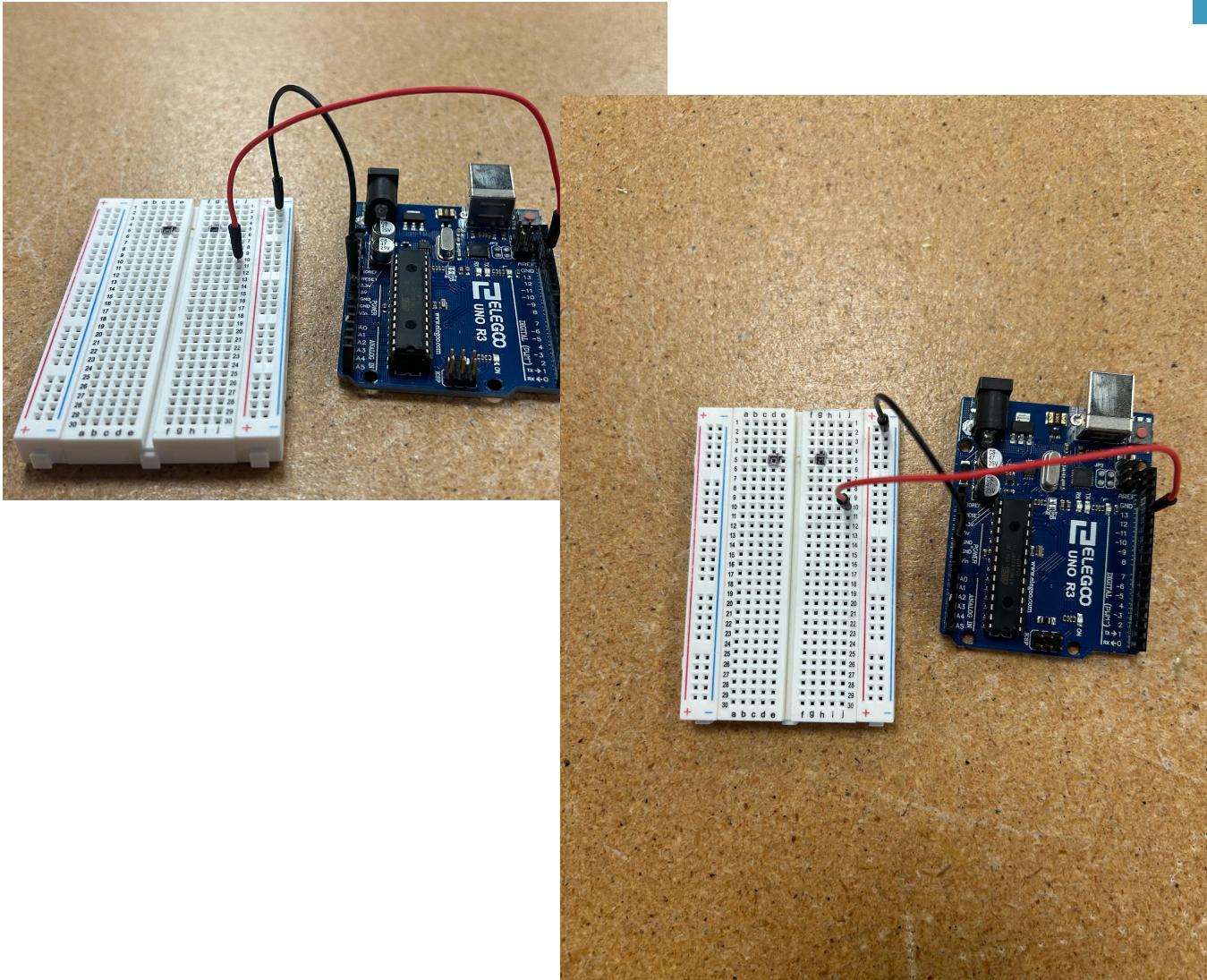


Figure 8: Breadboard Circuit Diagram for NOT Gate Using IC-7400

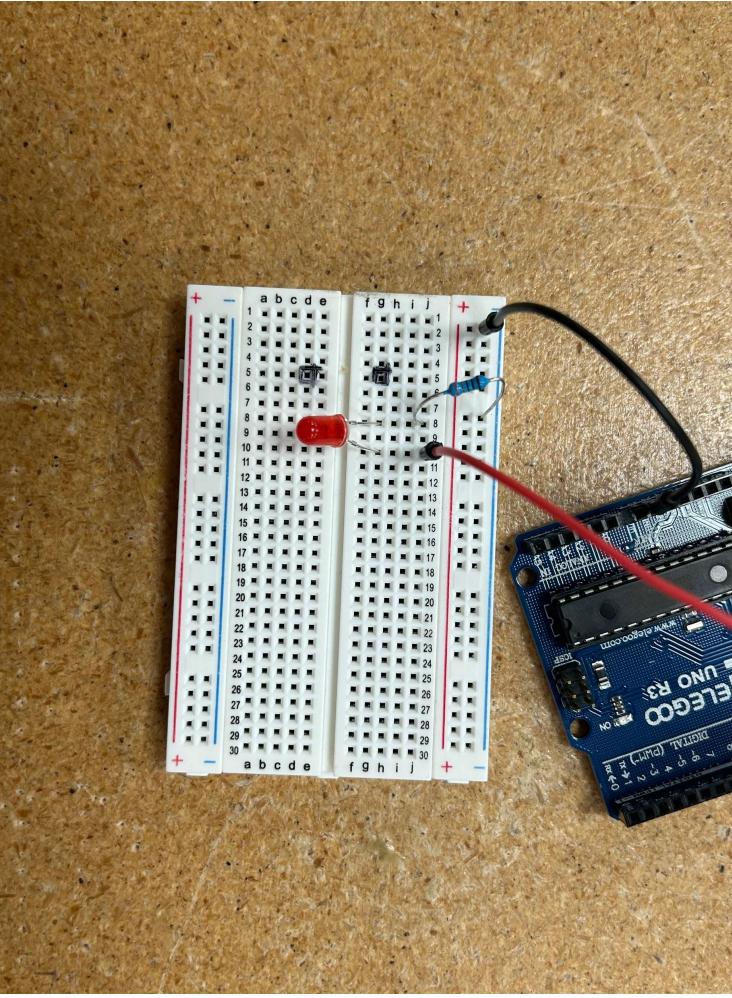
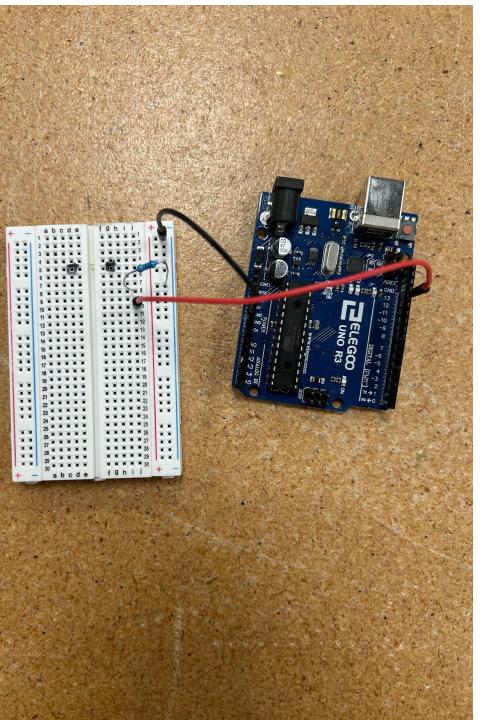
- Breadboards are used for prototyping and projects not meant to stay long term
- Parts can be moved easily and interchanged
- Breadboards electrically connect VERTICALLY for the GND (blue) line and for the PWR (red) line - the lines are not connected on opposite sides of the board
- The middle rows are connected ACROSS but the middle separates them
- Ex. Everything in 1 half of row 10 is electrically connected
- Ex. Everything in the first blue bus is connected but not to the second blue bus

Step 2: Make Connections



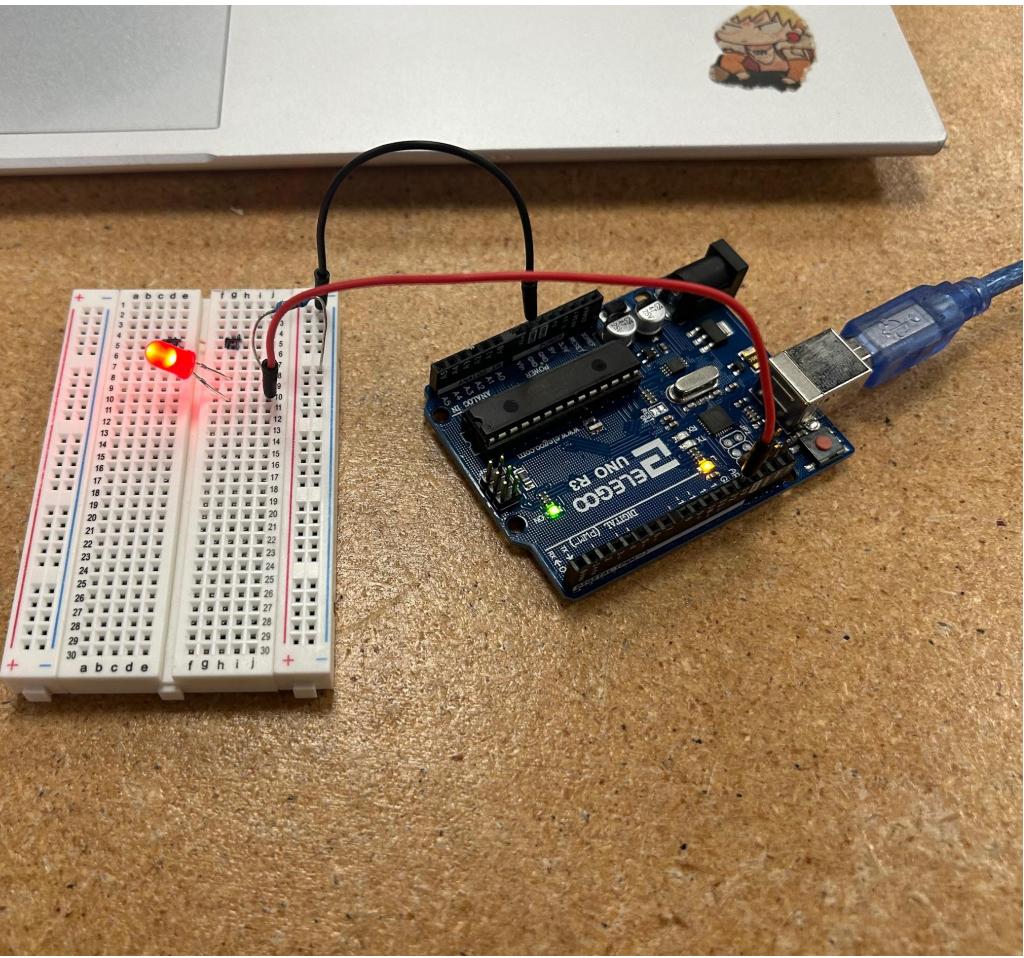
1. Connect 1 piece of hookup wire to the **GND** pin on the Arduino and the **blue bus** on the breadboard
2. Connect a second piece of wire to **Pin 13** in the Arduino board on the digital side into **row 10** on the breadboard

Step 2: Make Connections Cont.



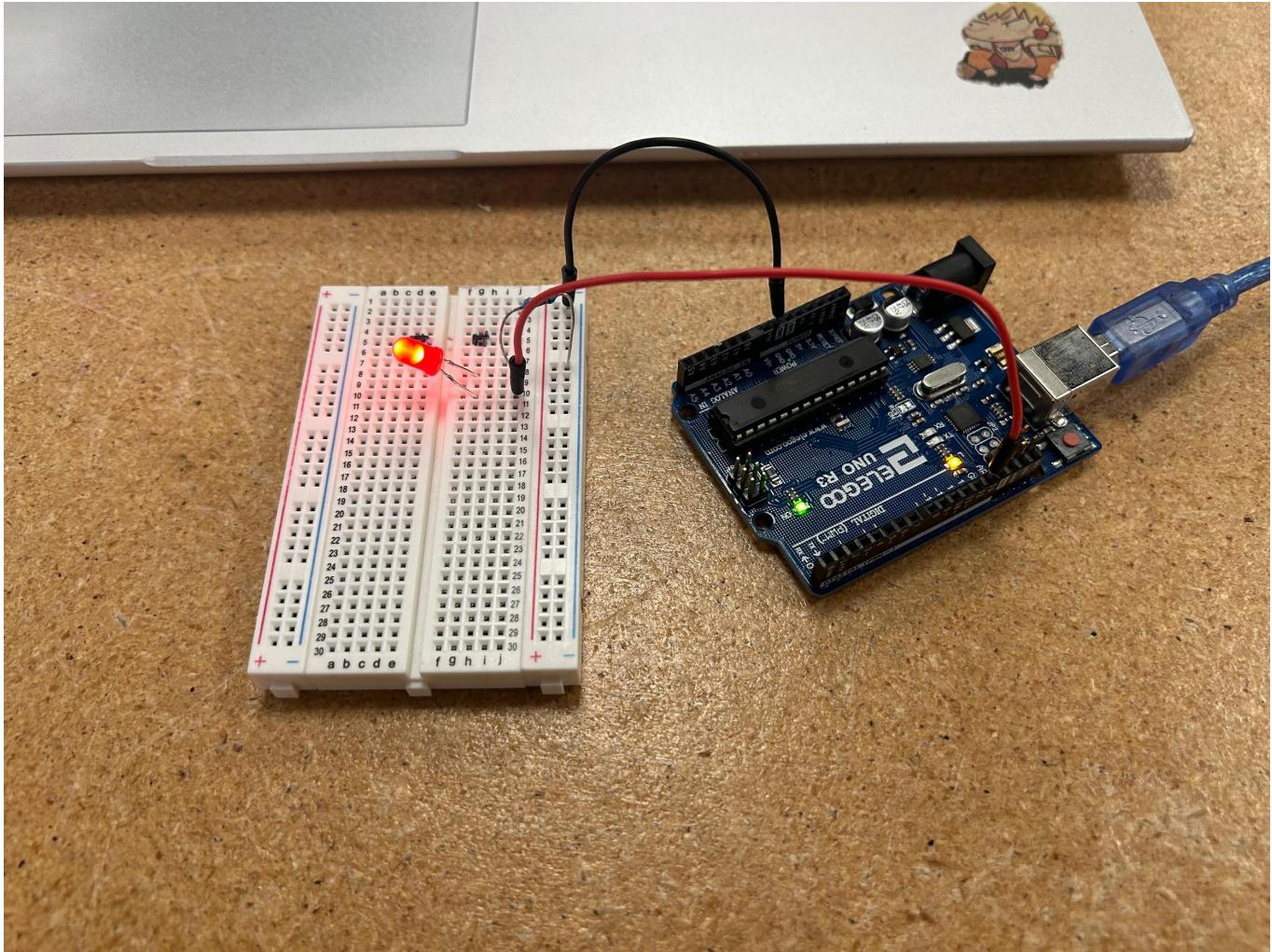
1. Connect the $470\ \Omega$ resistor (Yellow - Violet - Brown - Golden) to the blue bus and to row 8
2. Connect the LONG leg of one LED to row 10 and the SHORT leg into row 8

Step 3: Plug it in



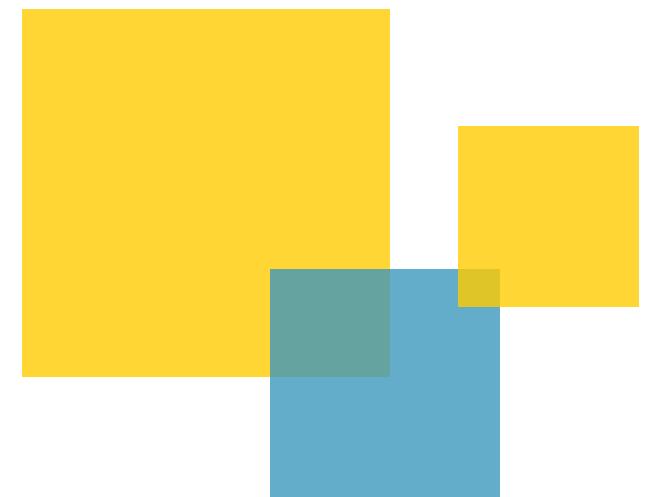
1. Connect the USB cord to the Arduino into the computer
2. Open the Arduino IDE
3. Add the blink code from the examples tab
File-> Examples-> Basics-> Blink
4. Check that the right port is selected
5. Click the checkmark button to verify the code
6. Click the arrow button to upload the code

Step 3: Plug it in Cont.



1. Check that the LED is blinking! If not check :
 - a. Is the LED in the right direction
 - b. Are the wires firmly planted in the holes
 - c. Are the wires in the correct buses and rows?
2. Congratulations on getting it to blink!

Experimentation Time



Step 1: Learn the Program

```
Blink.ino
1  /*
2   * Blink
3
4   * Turns an LED on for one second, then off for one second, repeatedly.
5
6   * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
7   * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
8   * the correct LED pin independent of which board is used.
9   * If you want to know what pin the on-board LED is connected to on your Arduino
10  model, check the Technical Specs of your board at:
11  https://www.arduino.cc/en/Main/Products
12
13  modified 8 May 2014
14  by Scott Fitzgerald
15  modified 2 Sep 2016
16  by Arturo Guadalupi
17  modified 8 Sep 2016
18  by Colby Newman
19
20  This example code is in the public domain.
21
22  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33     digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
34     delay(1000);                      // wait for a second
35     digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
36     delay(1000);                      // wait for a second
37 }
38 }
```

1. Code in between the /* and */ and // are comments and do not impact the program
2. In the setup function (ran once)
 - a. pinMode(LED_BUILTIN, OUTPUT);
 - i. The built in LED is setup using pinMode function.
3. In the loop function (repeats forever)
 - a. digitalWrite(LED_BUILTIN, HIGH);
 - i. Turns the LED on
 - b. delay(1000);
 - i. Wait 1 second
 - c. digitalWrite(LED_BUILTIN, LOW);
 - i. Turns the LED off
 - d. delay(1000);
 - i. Wait 1 second

Step 2: Edit the Program

1. Change the setup function of the program
 - a. Add multiple digital pins to the program

11-13

...

```
pinMode(13, OUTPUT);  
pinMode(12, OUTPUT);  
pinMode(11, OUTPUT);  
...
```

```
5 // the setup function runs once when you press reset or power the board  
6 void setup() {  
7     // initialize digital pin LED_BUILTIN as an output.  
8     pinMode(13, OUTPUT);  
9     pinMode(12, OUTPUT);  
10    pinMode(11, OUTPUT);  
11 }
```

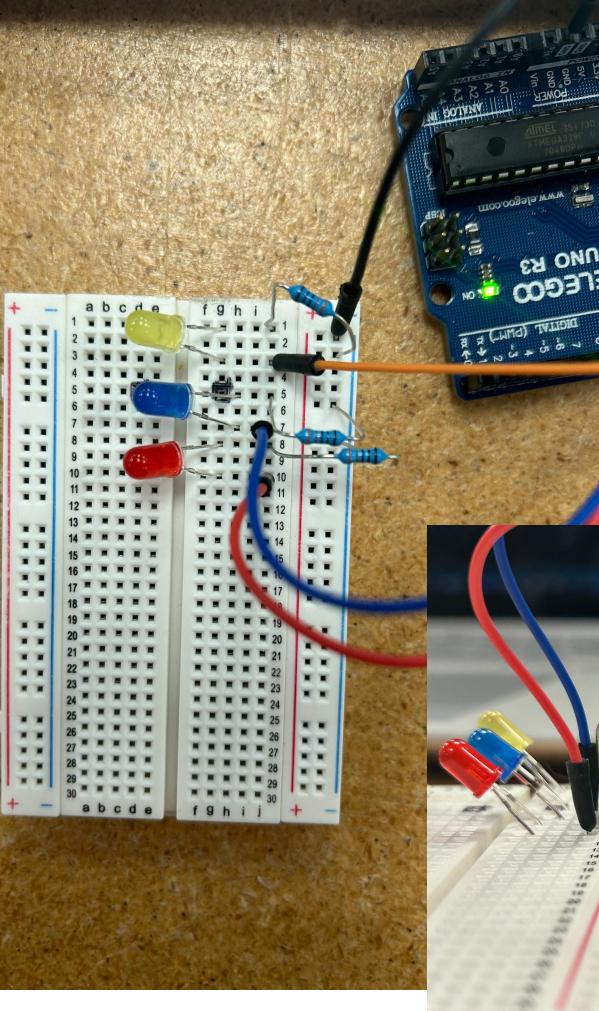
Step 2: Edit the Program Cont.

1. Change the loop function of the program
 - a. Get each of the pins to turn on at different times ...

```
13 // the loop function runs over and over again forever
14 void loop() {
15     digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
16     delay(1000);           // wait for a second
17     digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
18     delay(1000);           // wait for a second
19     digitalWrite(12, HIGH); // turn the LED on (HIGH is the voltage level)
20     delay(1000);           // wait for a second
21     digitalWrite(12, LOW); // turn the LED off by making the voltage LOW
22     delay(1000);
23     digitalWrite(11, HIGH); // turn the LED on (HIGH is the voltage level)
24     delay(1000);           // wait for a second
25     digitalWrite(11, LOW); // turn the LED off by making the voltage LOW
26     delay(1000);
27 }
```

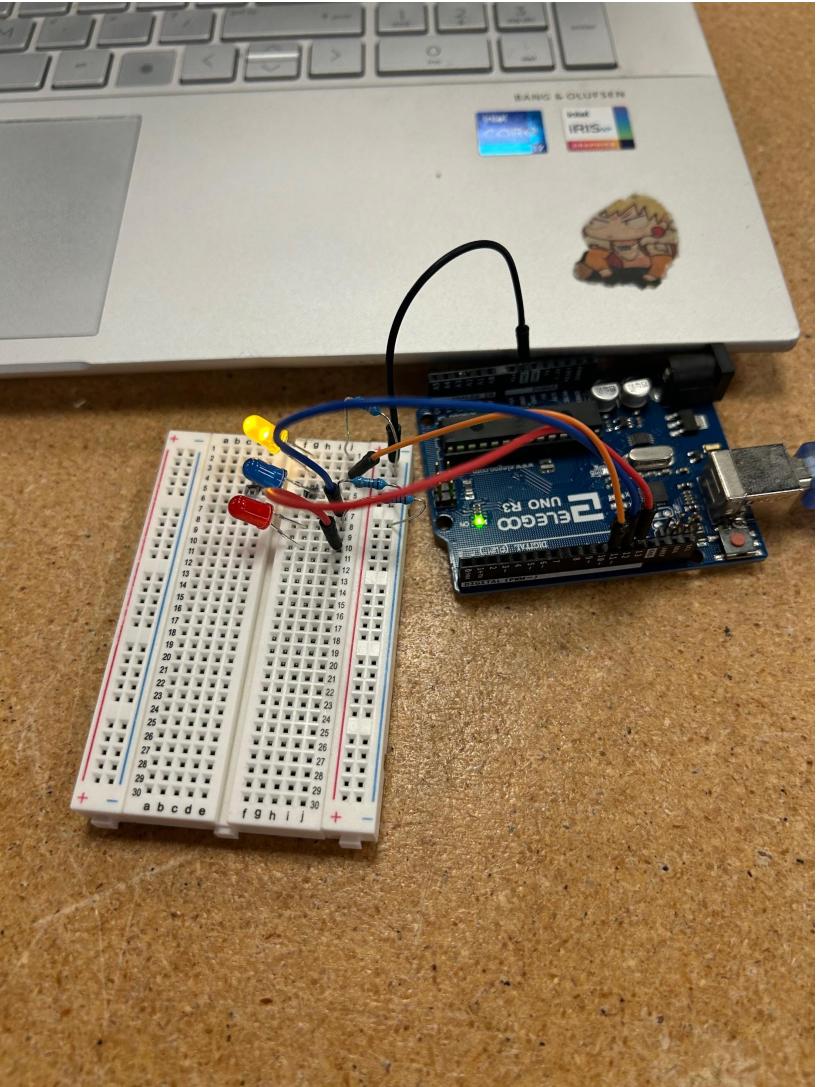
```
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
digitalWrite(12, HIGH);
delay(1000);
digitalWrite(12, LOW);
delay(1000);
digitalWrite(11, HIGH);
delay(1000);
digitalWrite(11, LOW);
delay(1000);
...
```

Step 3: Add More Lights



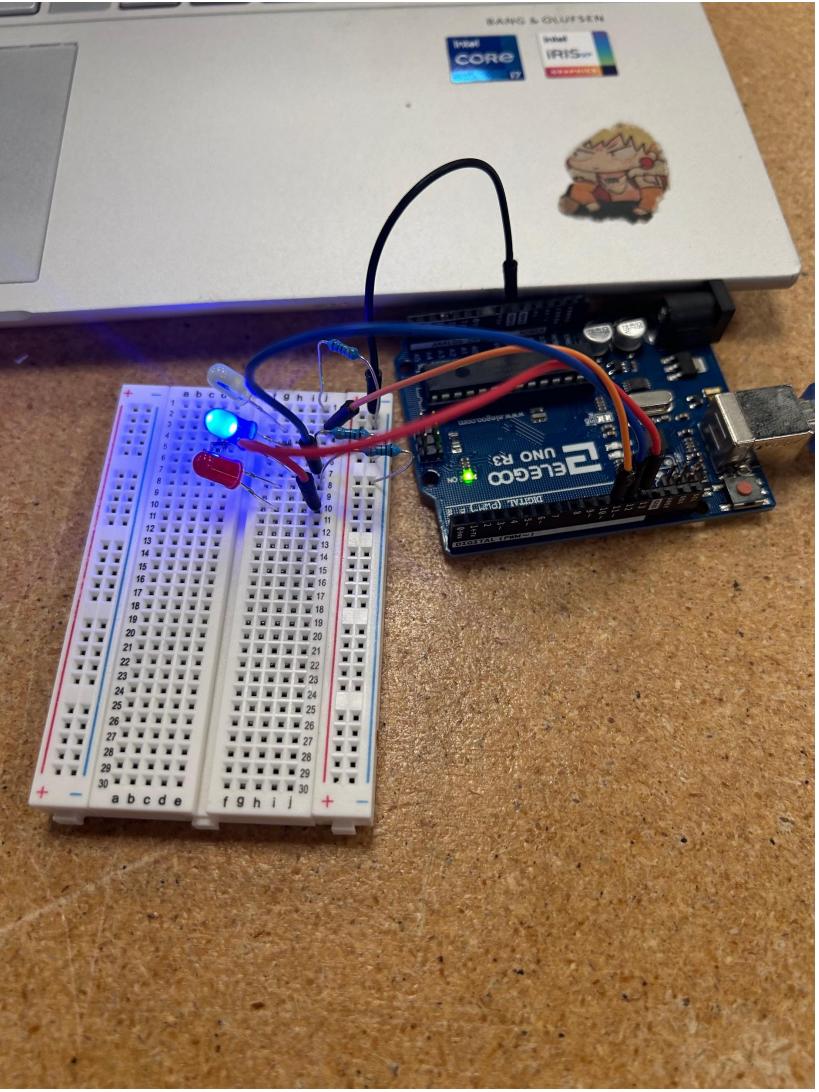
1. Add an LED connected to Pin 12
 - a. Use a piece of wire from pin 12 to row 7 in the breadboard
 - b. Connect the LONG end of the LED to row 7 and the short leg into row 5.
 - c. Connect the 470 Ω resistor (Yellow - Violet - Brown - Golden) to row 5 and the other end into the blue bus
2. Add an LED connect to Pin 11
 - a. Use a piece of wire from pin 11 to row 3 in the breadboard
 - b. Connect the LONG end of the LED to row 3 and the short leg into row 1.
 - c. Connect the 470 Ω resistor (Yellow - Violet - Brown - Golden) to row 1 and the other end into the blue bus

Step 3: Plug it in



1. Connect the USB cord to the Arduino into the computer
2. Open the Arduino IDE
3. Check that the right port is selected
4. Click the checkmark button to verify the code
5. Click the arrow button to upload the code

Step 3: Plug it in Cont.



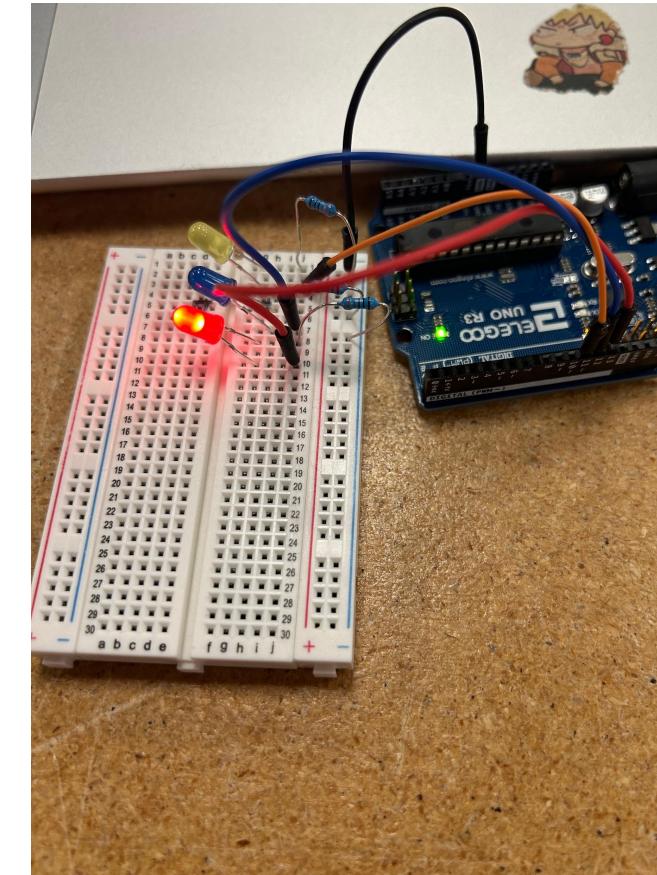
1. Check that the LEDs are blinking! If not check :
 - a. Is the LED in the right direction
 - b. Are the wires firmly planted in the holes
 - c. Are the wires in the correct buses and rows?
2. Congratulations on getting it to blink!

Step 4: Play with the Code

1. Personalize the code by changing different features about it!
2. Try changing delay(1000); to be more or less than 1000
3. Try changing the order of digitalWrite(13, HIGH); and
digitalWrite(13, LOW);
 - a. Remember the number corresponds to the pin number
 - b. HIGH means on
 - c. LOW means off
4. Remember to end every new line with a semicolon ;
5. Can you add a 4th LED to the system using the same pattern?
Don't forget to add it to the setup.

- What happens if the delay function is set to 500? What about 3000?
- Can multiple LEDs be turned on at the same time? How?
- What happens if the wire and LED is connected to a pin that is not included in the code? Why?

Questions



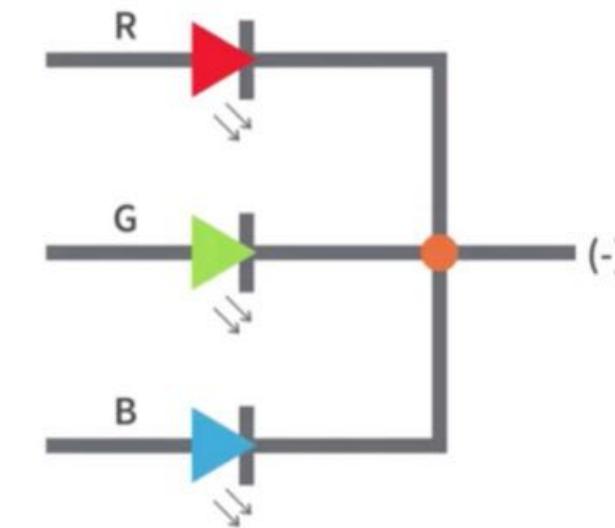
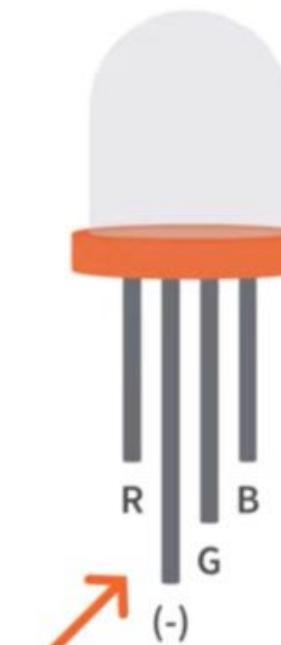


Arduino RGB

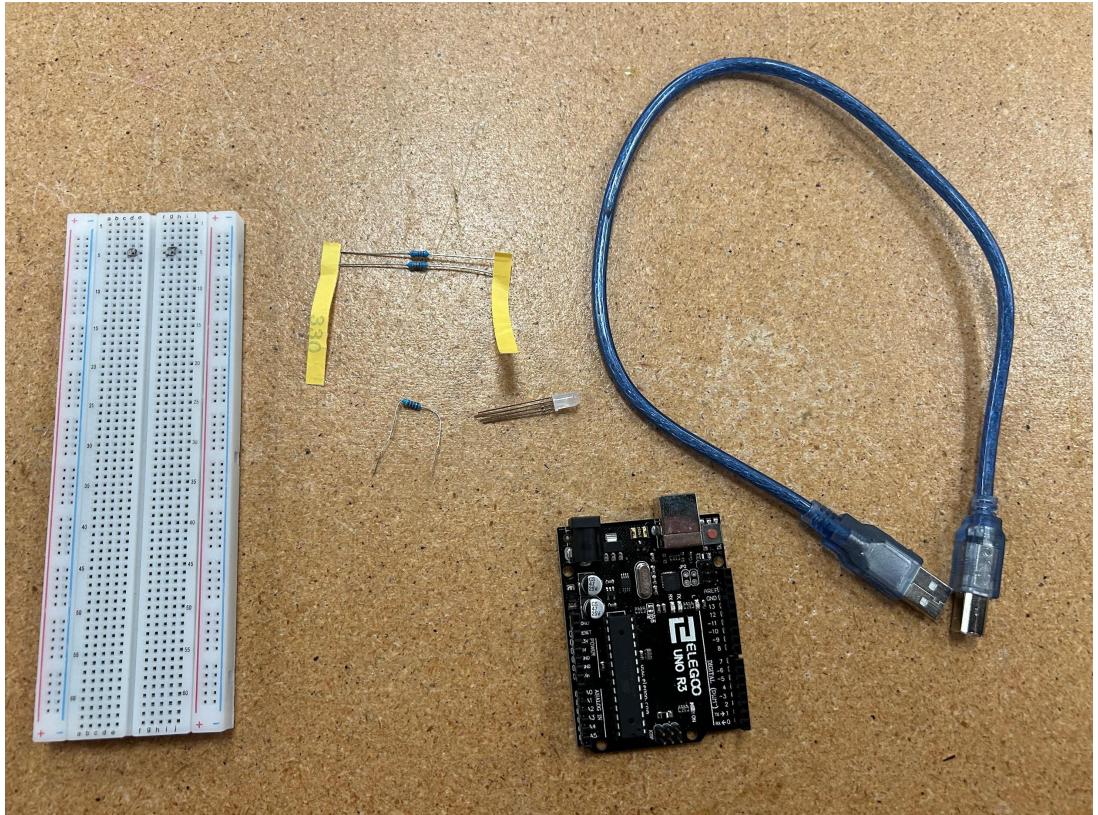


- 4 Pin LED
- Has 3 colors inside of it
Red - Green - Blue
- Each color can be controlled independently to create a wide range of colors
 - Just like the colors we see on a computer screen!
- The long leg is shared by all the LEDs and connected to GND
- The other legs correspond to a different color

RGB LED

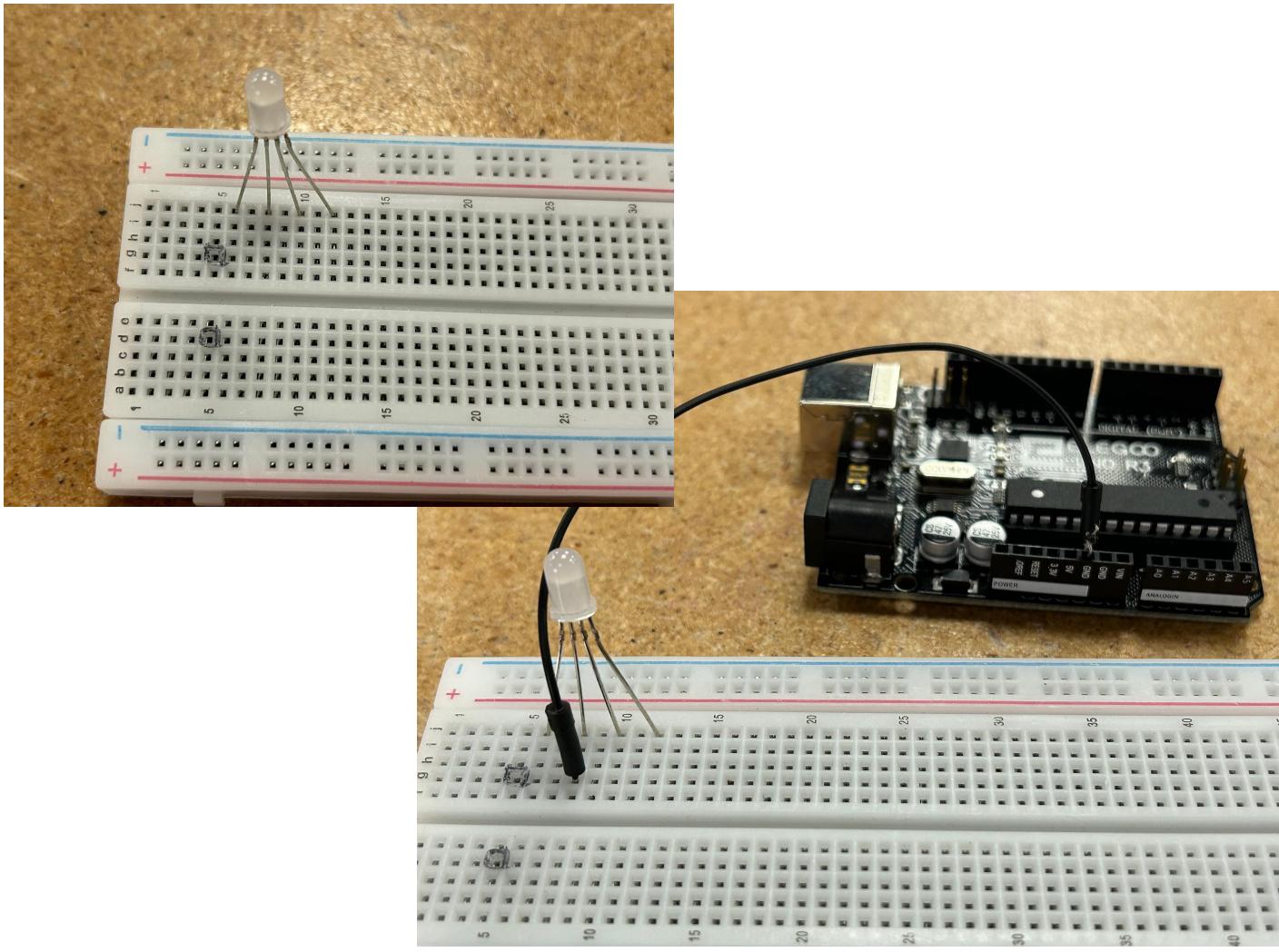


Step 1: Gather Materials



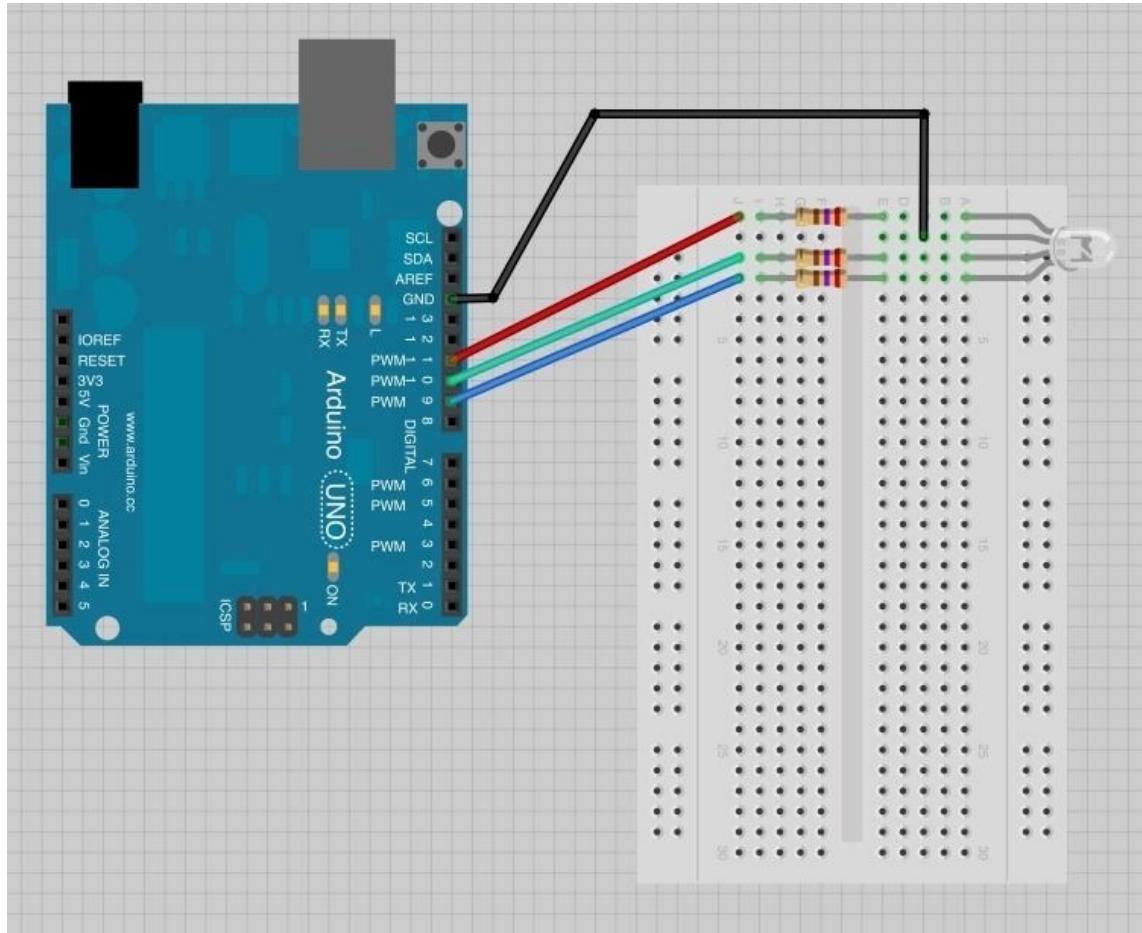
- 1 Arduino UNO
- 1 Breadboard
- 1 Common anode RGB LED
- 3 470 Ω resistor (Yellow - Violet - Brown - Golden)
- Hookup wire
- 1 Arduino USB cord

Step 2: Make Connections



1. Remove the previous work so you have a clean board
2. Put the RGB LED into the board with each leg in a different row (note which leg is in each row)
3. Connect 1 piece of hookup wire to to the GND pin on the Arduino and the LONG leg of the LED

Step 2: Make Connections Cont.



1. Connect a $470\ \Omega$ resistor (Yellow - Violet - Brown - Golden) to each of the other legs and the other side onto the other side of the breadboard.
2. Connect the red leg to digital pin 11
3. Connect the green leg to digital pin 10
4. Connect the blue leg to digital pin 9

Step 3: Plug it in

```
12 void setup()
13 {
14     pinMode(11, OUTPUT); // Red Pin
15     pinMode(10, OUTPUT); // Green Pin
16     pinMode(9, OUTPUT); // Blue Pin
17 }
18
```

1. Connect the USB cord to the Arduino into the computer
2. Open the Arduino IDE
3. Check that the right port is selected
4. Add the setup code to the file
...

```
pinMode(11, OUTPUT); // Red Pin
pinMode(10, OUTPUT); // Green Pin
pinMode(9, OUTPUT); // Blue Pin
...
```

Step 3: Plug it in Cont.

```
void loop()
{
    setColor(255, 0, 0); // red
    delay(1000);
    setColor(0, 255, 0); // green
    delay(1000);
    setColor(0, 0, 255); // blue
    delay(1000);
    setColor(255, 255, 0); // yellow
    delay(1000);
    setColor(80, 0, 80); // purple
    delay(1000);
    setColor(0, 255, 255); // aqua
    delay(1000);
}
```

1. Add the loop code to the file

```
...
void loop() {
    setColor(255, 0, 0); // Red Color
    delay(1000);
    setColor(0, 255, 0); // Green Color
    delay(1000);
    setColor(0, 0, 255); // Blue Color
    delay(1000);
    setColor(255, 255, 255); // White Color
    delay(1000);
    setColor(170, 0, 255); // Purple Color
    delay(1000);
    setColor(127, 127, 127); // Light Blue
    delay(1000);
}
...
```

Step 3: Plug it in Cont.

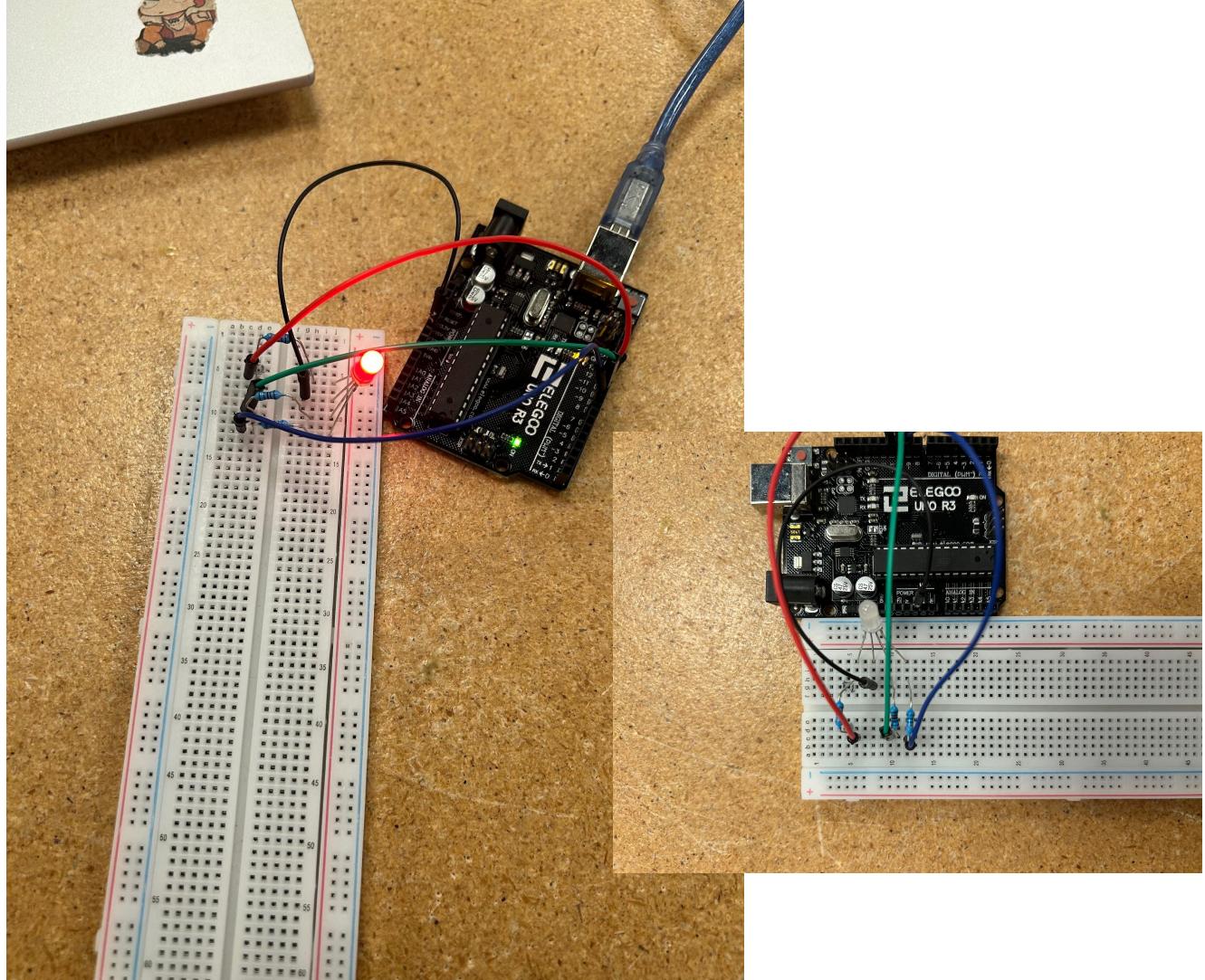
```
8 void loop() {  
9     setColor(255, 0, 0); // Red Color  
10    delay(1000);  
11    setColor(0, 255, 0); // Green Color  
12    delay(1000);  
13    setColor(0, 0, 255); // Blue Color  
14    delay(1000);  
15    setColor(255, 255, 255); // White Color  
16    delay(1000);  
17    setColor(170, 0, 255); // Purple Color  
18    delay(1000);  
19    setColor(127, 127, 127); // Light Blue  
20    delay(1000);  
21 }  
22  
23 void setColor(int redValue, int greenValue, int blueValue) {  
24     analogWrite(11, redValue);  
25     analogWrite(10, greenValue);  
26     analogWrite(9, blueValue);  
27 }  
28
```

1. After the loop code create a new function called setColor!
2. Remember to end with a }

```
...  
  
void setColor(int redValue, int greenValue, int  
blueValue) {  
    analogWrite(11, redValue);  
    analogWrite(10, greenValue);  
    analogWrite(9, blueValue);  
}  
...
```

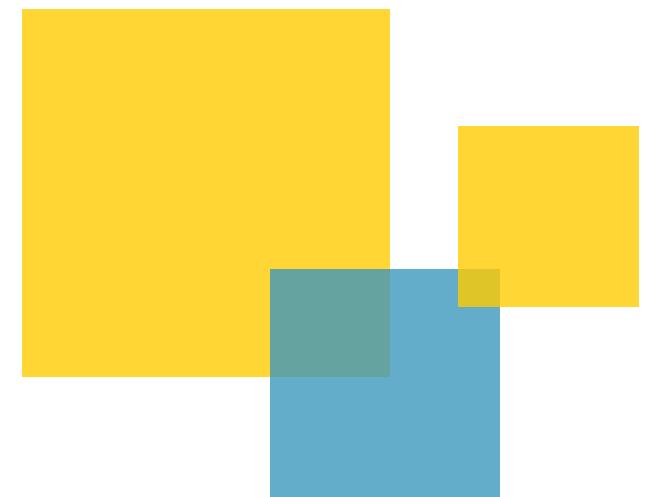
3. Click the checkmark button to verify the code
4. Click the arrow button to upload the code

Step 3: Plug it in Cont.



1. Check that the LED is changing colors! If not check :
 - a. Is the LED in the right direction? Are the legs correct?
 - b. Are the wires firmly planted in the holes
 - c. Are the wires in the correct buses and rows?
2. Congratulations on getting it to change colors!

Experimentation Time



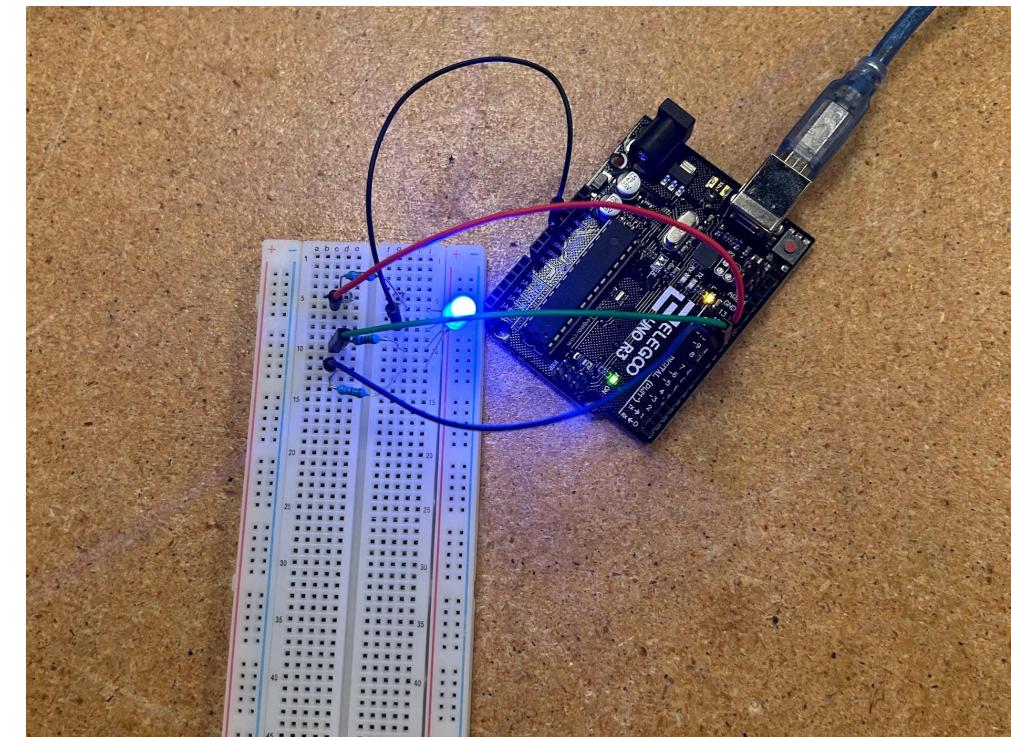
Step 1: Learn the Program

```
1 void setup()
2 {
3     pinMode(11, OUTPUT); // Red Pin
4     pinMode(10, OUTPUT); // Green Pin
5     pinMode(9, OUTPUT); // Blue Pin
6 }
7
8 void loop() {
9     setColor(255, 0, 0); // Red Color
10    delay(1000);
11    setColor(0, 255, 0); // Green Color
12    delay(1000);
13    setColor(0, 0, 255); // Blue Color
14    delay(1000);
15    setColor(255, 255, 255); // White Color
16    delay(1000);
17    setColor(170, 0, 255); // Purple Color
18    delay(1000);
19    setColor(127, 127, 127); // Light Blue
20    delay(1000);
21 }
22
23 void setColor(int redValue, int greenValue, int blueValue) {
24     analogWrite(11, redValue);
25     analogWrite(10, greenValue);
26     analogWrite(9, blueValue);
27 }
28 }
```

1. Code in between the /* and */ and // are comments and do not impact the program
2. In the setup function (ran once)
 - a. pinMode(11, OUTPUT); // Red Pin
 - i. Sets up each pin and its corresponding color
3. In the loop function (repeats forever)
 - a. setColor(170, 0, 255); // Purple Color
 - i. Turns the LED on
 - ii. Sets the color to purple
 - iii. Each of the numbers represents a color value with 0 being the lowest and 255 being the highest
 - iv. Red value 170 so over half
 - v. Green value at 0 so none
 - vi. Blue value at 255 so completely on

- What happens if the delay function is set to 500? What about 3000?
- How do you make yellow? How is it possible?
- What other colors can you make using the number system? Hint: Use an online RGB color picker from google.
- Can you make it go through the rainbow?

Questions



Step 2: Edit the Program

```
1 int red = 11;
2 int green = 10;
3 int blue = 9;
4 int t = 10; // Time for delay
5
6 void setup(){
7     pinMode(red, OUTPUT);
8     pinMode(green, OUTPUT);
9     pinMode(blue, OUTPUT);
10}
11 void Fade(){
12    for( int i = 0; i <= 255; i++){
13        analogWrite(red, i); // fade up
14        analogWrite(green, 255 - i); // fade down
15        analogWrite(blue, 0); // do nothing
16        delay(t);
17    }
18    for( int i = 0; i <= 255; i++){
19        analogWrite(red, 255 - i); // fade down
20        analogWrite(green, 0); // do nothing
21        analogWrite(blue, i); // fade up
22        delay(t);
23    }
24    for( int i = 0; i <= 255; i++){
25        analogWrite(red, 0); // do nothing
26        analogWrite(green, i); // fade up
27        analogWrite(blue, 255 - i); // fade down
28        delay(t);
29    }
30}
31 void loop(){
32    Fade();
33}
34|
```

1. Make the colors fade through the rainbow
2. Add these variables and setup
```

int red = 11;  
int green = 10;  
int blue = 9;  
int t = 10; // Time for delay

```
void setup(){
 pinMode(red, OUTPUT);
 pinMode(green, OUTPUT);
 pinMode(blue, OUTPUT);
}
```
```

Step 2: Edit the Program Cont.

```
1 int red = 11;
2 int green = 10;
3 int blue = 9;
4 int t = 10; // Time for delay
5
6 void setup(){
7     pinMode(red, OUTPUT);
8     pinMode(green, OUTPUT);
9     pinMode(blue, OUTPUT);
10}
11 void Fade(){
12     for( int i = 0; i <= 255; i++){
13         analogWrite(red, i); // fade up
14         analogWrite(green, 255 - i); // fade down
15         analogWrite(blue, 0); // do nothing
16         delay(t);
17     }
18     for( int i = 0; i <= 255; i++){
19         analogWrite(red, 255 - i); // fade down
20         analogWrite(green, 0); // do nothing
21         analogWrite(blue, i); // fade up
22         delay(t);
23     }
24     for( int i = 0; i <= 255; i++){
25         analogWrite(red, 0); // do nothing
26         analogWrite(green, i); // fade up
27         analogWrite(blue, 255 - i); // fade down
28         delay(t);
29     }
30 }
31 void loop(){
32     Fade();
33 }
34 |
```

1. Make the Fade function. It gradually increases and decreases in values of red, green, and blue

...

```
void Fade(){
    for( int i = 0; i <= 255; i++){
        analogWrite(red, i); // fade up
        analogWrite(green, 255 - i); // fade down
        analogWrite(blue, 0); // do nothing
        delay(t);
    }
    for( int i = 0; i <= 255; i++){
        analogWrite(red, 255 - i); // fade down
        analogWrite(green, 0); // do nothing
        analogWrite(blue, i); // fade up
        delay(t);
    }
    for( int i = 0; i <= 255; i++){
        analogWrite(red, 0); // do nothing
        analogWrite(green, i); // fade up
        analogWrite(blue, 255 - i); // fade down
        delay(t);
    }
}
```

...

Step 2: Edit the Program Cont.

```
1 int red = 11;
2 int green = 10;
3 int blue = 9;
4 int t = 10; // Time for delay
5
6 void setup(){
7     pinMode(red, OUTPUT);
8     pinMode(green, OUTPUT);
9     pinMode(blue, OUTPUT);
10}
11 void Fade(){
12    for( int i = 0; i <= 255; i++){
13        analogWrite(red, i); // fade up
14        analogWrite(green, 255 - i); // fade down
15        analogWrite(blue, 0); // do nothing
16        delay(t);
17    }
18    for( int i = 0; i <= 255; i++){
19        analogWrite(red, 255 - i); // fade down
20        analogWrite(green, 0); // do nothing
21        analogWrite(blue, i); // fade up
22        delay(t);
23    }
24    for( int i = 0; i <= 255; i++){
25        analogWrite(red, 0); // do nothing
26        analogWrite(green, i); // fade up
27        analogWrite(blue, 255 - i); // fade down
28        delay(t);
29    }
30}
31 void loop(){
32     Fade();
33 }
34
```

1. Finish the loop function

...

```
void loop(){
Fade();
}
```

2. Check that the right port is selected
3. Click the checkmark button to verify the code
4. Click the arrow button to upload the code
5. Check that the LED is changing colors! If not check :
 - a. Is the LED in the right direction? Are the legs correct?
 - b. Are the wires firmly planted in the holes
 - c. Are the wires in the correct buses and rows?
6. Congratulations on getting it to change colors! Isn't it pretty!

*Congratulations on
Completing your first
Arduino coding projects!*