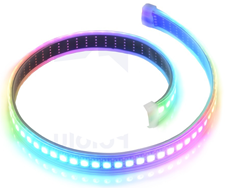
**Creative Programming With LEDs**

Girls Inc. Eureka Workshop – U. Massachusetts July 2017

**Part 1: Build it!**

**Objective:** Building LED art often requires setting up custom hardware. For this lab we will use Arduino Uno and AdaFruit Neopixel, which needs to be connected as follows. The **green wire** will need to connect to 2, the **black wire** need to connect the GND pin (one of the two GNDs), and the **red wire** needs to connect to 5V

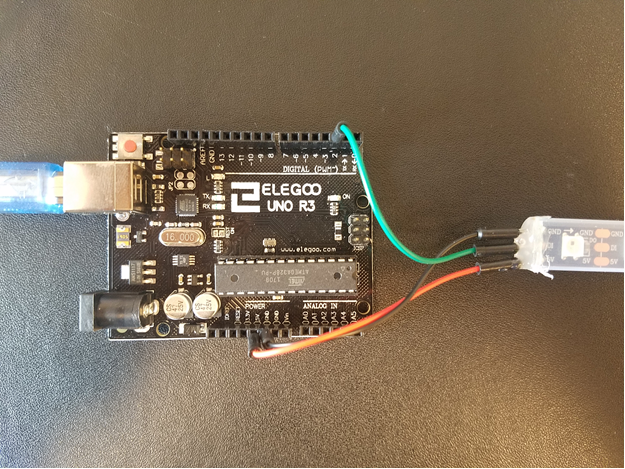


Arduino Uno

AdaFruit

NeoPixel

Your connections should look something like the following.



**Part 2: Getting familiar with the LEDs**

**Objective:** The objective of this part of the activity is to become familiar with running code and making changes to existing code. You will learn how to change the LED’s color and location.

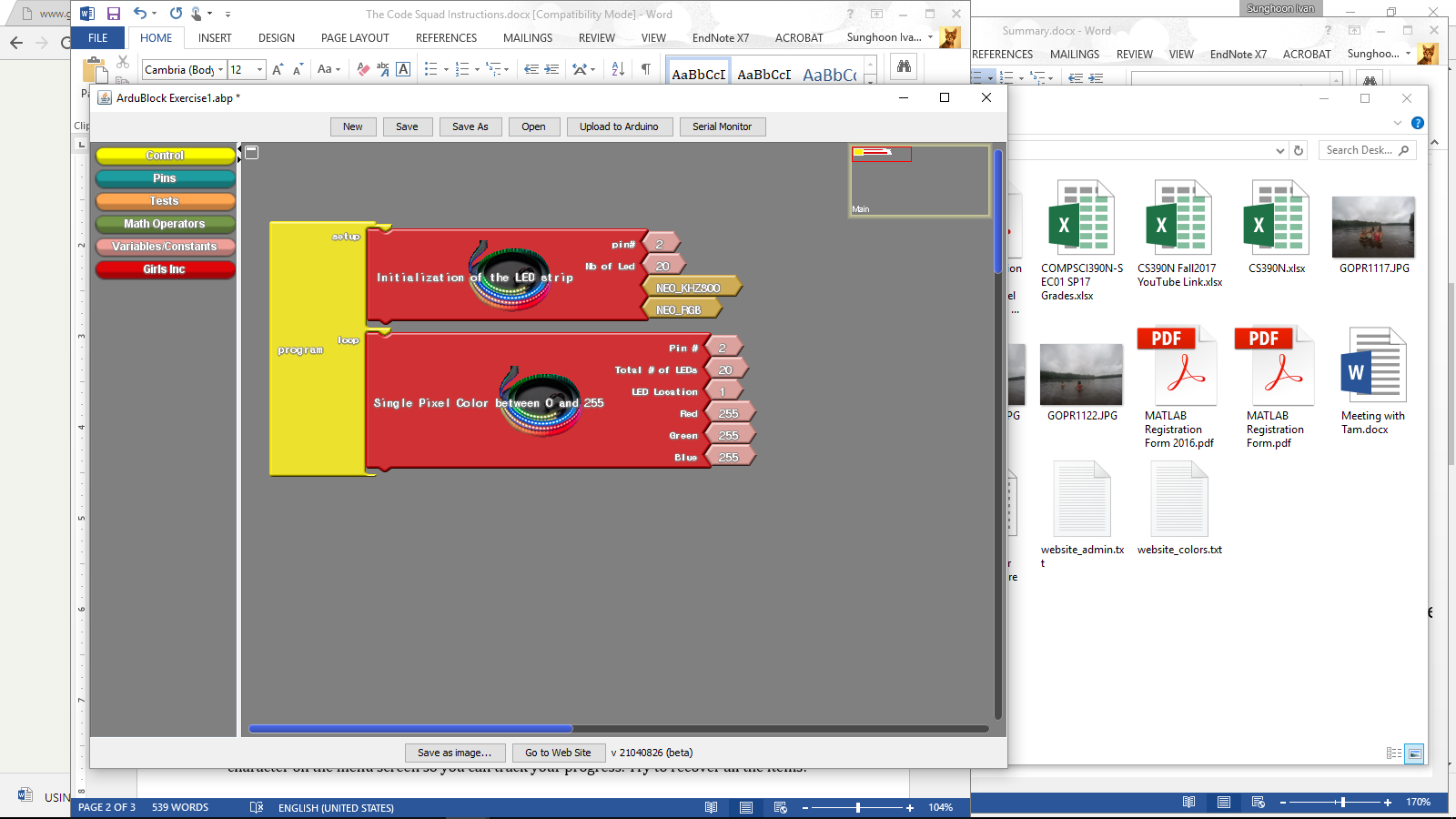
**Activity 1: Run your first program!**

* Navigate to the Labs folder.
* Open the Lab 1 folder.
* Double click on “Exercise1.abp”

**Welcome to your first Arduino program!**



To run your program click “Upload to Arduino” in this menu that is above the sketch.



Based on what the LEDs are doing can you guess how the **Only One Pixel Color** block is working?

Can you modify the code to make the LED change color?

**Activity 2: Experiment with color**

Color is specified using 3 values representing the amount of red, green and blue.

What happens if you specify larger numbers for each color (e.g., set them all to a given value)? Try: 255, 127, 23, 0.

Can you make the LED purple? Green? Yellow?

**Activity 3: Light up more LEDs**

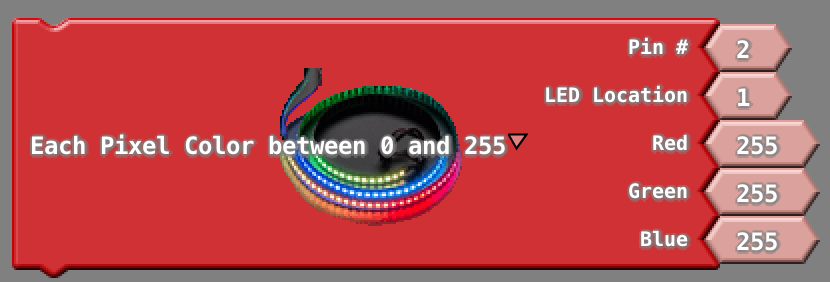
Can you modify your program to light up an LED at a different location?

Try lighting up LEDs at different positions on your strip

What is the smallest value you can specify for position? The largest? What happens if you give invalid input? Try using multiple of the **Only One Pixel Color** block with different input values. What does it do?



Try the same thing using the **Each Pixel Color** block (in the **Girls Inc** menu)

****

**Part 3: Telling the Arduino what to do**

**Objective:**  In this activity, you will learn how to modify the code to have the program perform multiple actions. You will learn about how we can specify what we want the LEDs to do.

**Activity 1: Have the LED change colors once per second**

Click on the **Girls Inc** menu. Select the “**Only One Pixel Color**” block and drag it into your loop. Repeat this to put multiple commands into the loop function. Try the example below:



Try running your code. What do you observe?

Click on the “**Control**” menu. Drag and drop the “**delay**” function into your loop. Experiment with different values of delay (e.g., 1,000 ms, 500 ms)



**Activity 2: Make the LED “move”**

Edit the parameters of the “**Only One Pixel Color**” function to “move” your LED 4 positions to the right (you may need to get more blocks from the **GirlsInc** menu for this).

Can you make it 4 positions to the left?

What about changing colors while it moves?

What happens if you insert the “Delay” command between the “**Only One Pixel Color**” functions?

**Part 4: Control structures, making your life easier!**

**Objective:** In the last exercise, you saw how issuing multiple commands can change the behavior of the LEDs. We will now see how programmers use loops and variables to make this easier.

**Activity 1: Using a loop to move the LED**

Under the yellow control menu, drag the “Repeat & Count” block. (You may want to remove any blocks within your loop before placing the new block).



This block allows you to repeat the command within the block a given number of times. It also allows you to keep a count of how many times it has executed using the variable, called “**variable name**”. The block above will repeat the items in “commands” 5 times.

**Using a variable:**

To use the “**variable name**” in your loop, select the “**Integer**” block from the “**Variables/Constants**” menu. It will look something like this:



Click on the “**Integer**” block and type “**variable name**”. This block will now carry the value of our variable called “**variable name**”. It will look like this:



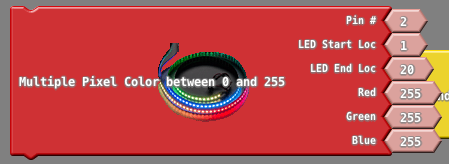
You now need to place your variable where you would like to use it. In the below example, we have dragged the value of **LED Location** out of the **Only One Pixel Color** block, and replaced it with our variable that we made in the previous step.



Try running the example above. What does the code do? Can you use the **delay** block to change the speed of this code? Can you change the color of the pixel in this example code?

**Activity 2: Color multiple pixels using a new function**

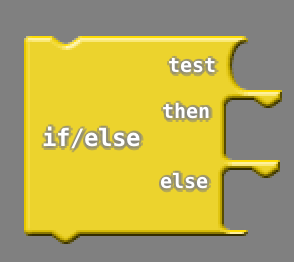
Under the **Girls Inc** menu, select the “**Multiple Pixel Color**” function.



Try using this function to turn all the LEDs a single color (e.g., Red, Purple, etc.). What happens if you change **LED Start Loc? LED End Loc?**

Now, use a loop and this new function to change colors of the LEDs in a specific sequence (e.g., Red🡪Green🡪Blue).

Here we will use an “**If/Else**” block from the **Control** menu. It looks like this:



If the condition in the “**test**” position is true, then the commands in the **then** block will be executed. Otherwise commands in the **else** block will be executed.

To form a **test** click on the **tests** category and select the equals test (as an example). It looks like the following:



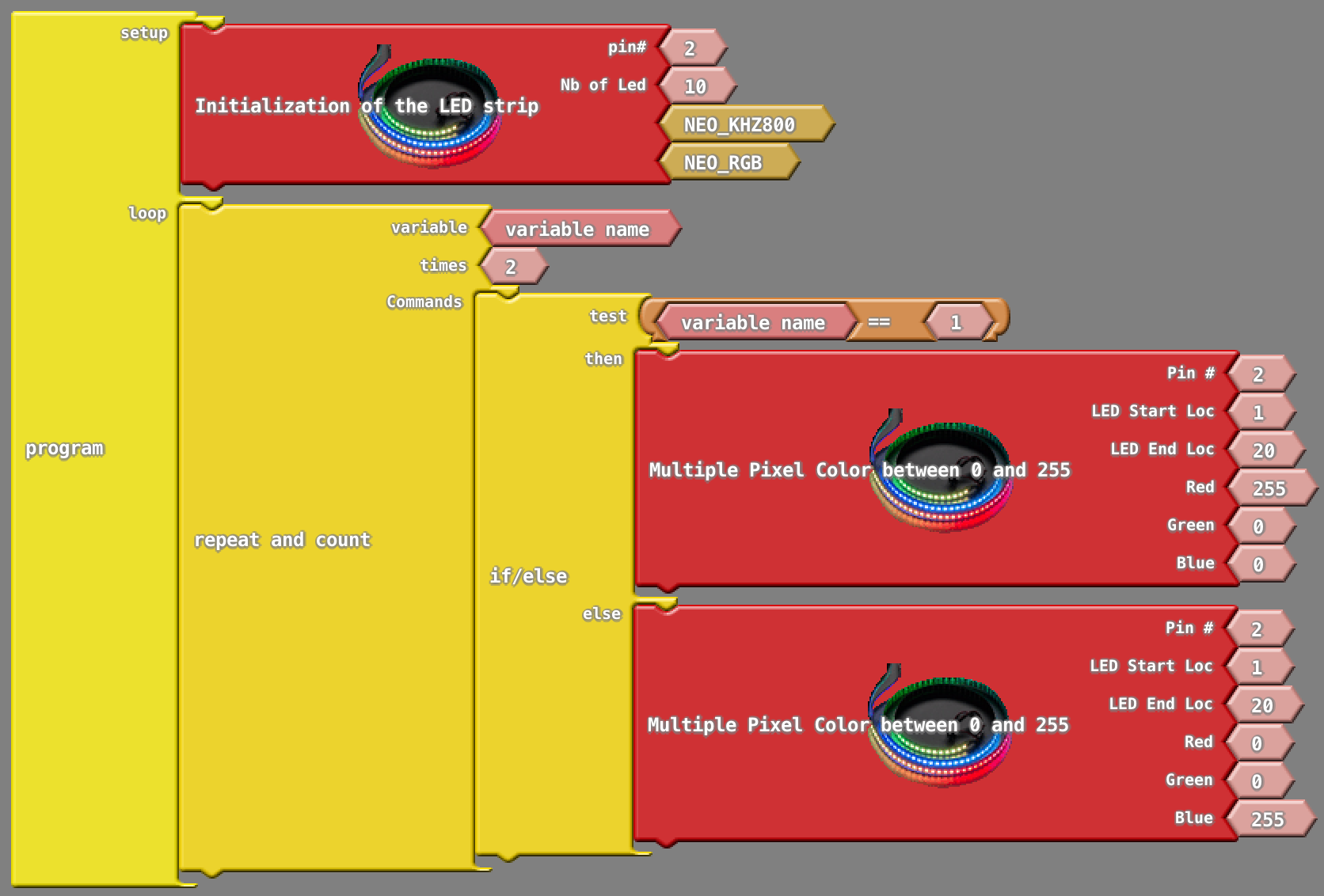
We can put a variable (like how we used “variable name” in the prior example) in one side of the test, and a constant value in the other side. This allows us to take a specific action if the variable has a specific value.

You can get a constant value by clicking on **Variables/Constants** and selecting the light pink block with a ‘1’ in it:



You can change the value of this constant by clicking and typing a different numerical value.

Try the example code below. Try reading the code to figure out what it is doing before you run it. It might help to write out the different steps it will take.



Can you update this code to alternate between three different colors? **Hint:** You will want to increase the **Repeat and count** to repeat 3 **times** . You may also consider adding another if-statement.

**Activity 3 (Challenging Mission): Using a loop to move the LED but with alternating colors**

In Activity 1, you made a code that moves the LED from one location to another. Now, why don’t you choose two favorite colors of yours that will be alternated throughout the LED sweep! For example, if your favorite colors and PINK and BLUE, you want to have something like:

Location #1 (pink) 🡪 Location #2 (blue) 🡪 Location #3 (pink) 🡪 … 🡪 Location #30 (blue)

If you successfully make this code, try different “delay” lengths and see what happens.

This is a challenging mission, but why don’t you give it a try!

**Hint**: you need a repeat-and-count block, an if-statement, and some math functions to detect odd/even numbers!

**Part 5: Adding An Additional Block!**

**Activity 1: Build a program that turn on LEDs (#1 - #20) with a color of your choice for 0.5 second, turn off for 0.5 seconds, and repeat…**

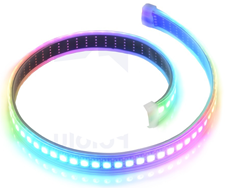
This shall be relatively easy to build compared to your previous missions. ☺ **Make sure that you are only turning on LEDs from location #1 to #20**.

**Hint**: You can put R = 0, G = 0, B = 0 to turn off the LED. You also need to use at least two “delay” blocks.

**Activity 2: Changing the delay with your knob sensor.**

Now that your sensor can periodically turn on/off every 0.5 seconds, why don’t we try to adjust the delay in real-time using our knob sensor?

First of all, let us connect the knob sensor as the following. The **white wire** will need to connect to A0, the **black wire** need to connect the GND pin (next to the LED’s GND pin), and the **red wire** need to connect to 3.3V.



Your connections shall look something like the following.



Now we need to make the code working! Go ahead and open up a new panel on the ArduBlock. Then,

1. Click on the “Pins” menu, select “analog pin #” block, and drag it into your drawing panel. This block will read values from your knob sensor, which is connect to A0.
2. Now try to plug the “analog pin #” into the “delay” block. (you should first remove the number that is connected to your “delay” block).



1. Turn your knob all the way to the clockwise, and start to slowly turn your knob counterclockwise. (WARNING: make sure to slowly turn, the knob is very sensitive!!) What do you see?!
2. Can you think about any other things that you can control with your knob? Feel free to explore!

