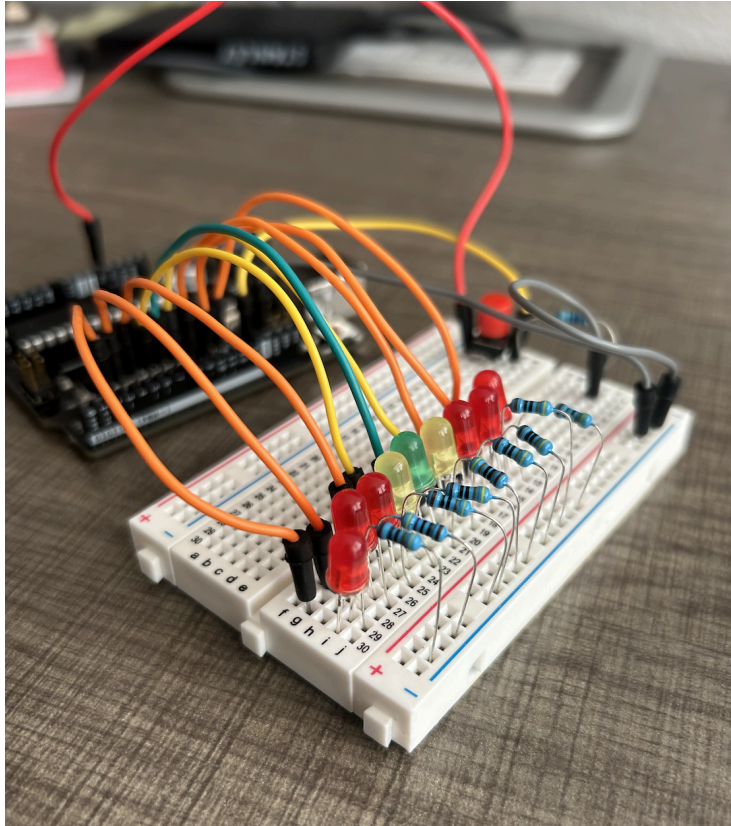


Stop the LED!

By Stanley Wong



Showcase / Description of Finished Piece

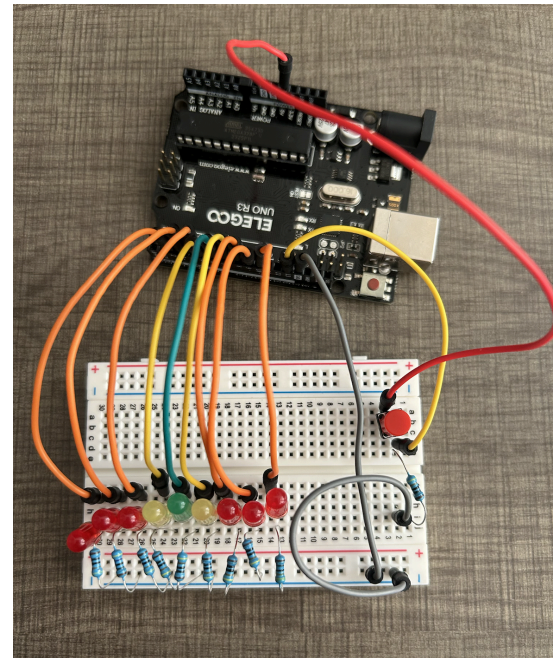


Taking inspiration from a classic arcade game (pictured to the left), I wanted to try recreating a mini version using what we learned during class about buttons and LEDs. The classic arcade game often involves several LEDs placed along the circumference of a large circular machine. The LEDs will continue to light up consecutively around the circle at a very fast pace. The ultimate goal of the game is the press a large button at a very specific time in the game to stop the

lights from moving around and to collect the indicated number of tickets for that particular light.

My project works in a very similar way to this type of game. Instead of the LEDs being placed in a circle, this project has the LEDs placed in a line and involves the LEDs lighting up back and forth along the LEDs so that when the lit up LED is the last in the row, it will reverse and light up in the reverse direction. The LEDs are color coded based on the idea that the "jackpot" would be located in the center, with point values decreasing as you move outwards.

As the LEDs are lighting up, there is a button on the right hand side. When that button is pressed and held, the corresponding LED that is lit up at the time that the button is pressed will remain lit while the others remain unlit. This allows for the user to see exactly where the LED has stopped.

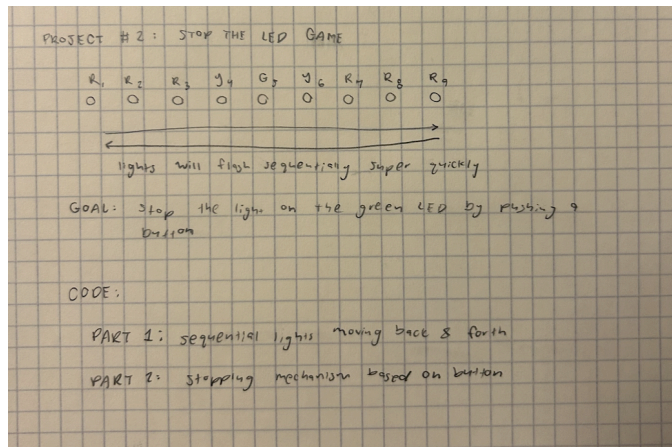


Linked below is a video that demonstrates how this game is played:

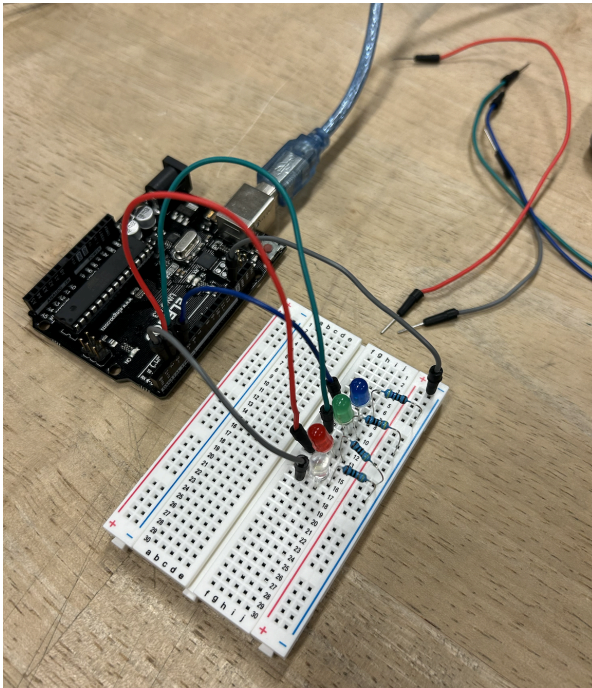
[Expressive Light Project: Stop the LED! Demo](#)

Process

Below is an image of my original sketch for my idea for this game.



There were 3 primary sections in my process in creating this game, all of which are expanded on below in depth.



```
//Define the first LED
int led = 2;

//Define button
const int ButtonPin = 13;

byte ButtonState = LOW;

void setup() {
  // initialize all pins as an output
  for (int led=2; led<6; led++){
    pinMode(led, OUTPUT);
  }

  // initialize button as an input
  pinMode(ButtonPin, INPUT);

  //all pins start off
  for (int led=2; led<6; led++){
    digitalWrite(led, LOW);
  }
}

void loop() {
  byte ButtonState = digitalRead(ButtonPin);

  if (ButtonState == HIGH){
    digitalWrite(led, HIGH);
  }

  else{
    // lights turn on & off sequentially from 1 -> 9
    for (int i=1; i<5; i++){
      digitalWrite(led, HIGH);
      delay(100);
      digitalWrite(led, LOW);
      led = led + 1;
    }

    // lights turn on & off sequentially from 9 -> 1
    for (int i=1; i<5; i++){
      digitalWrite(led, HIGH);
      delay(100);
      digitalWrite(led, LOW);
      led = led - 1;
    }
  }
}
```


Part 1: LED Chasing[Expressive Light Project: LED Chasing](#)[Stop the LED: Version 1 Code](#)

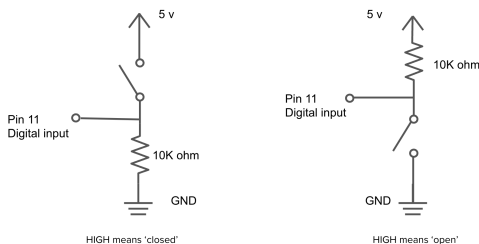
The first hurdle I needed to figure out was to figure out how I was going to (1) wire the LEDs together to form a line structure which would allow it to light up quickly back and forth and (2) code a loop that would allow this to happen. I decided to start with just four LEDs so that the wiring and code wouldn't be too overwhelming at first, which I would then be able to build out later. Fortunately, there were many sample wiring diagrams and sample code on the internet that I was able to reference. All I needed to do was to adjust it specifically to the needs of my project.

References:

- ▶ Arduino Sequential LED Chaser with 12 Effects | Arduino LED Chaser with Code
- ▶ Arduino Tutorial: LED Sequential Control- Beginner Project
- ▶ Arduino sequential LED project | led chaser circuit | arduino project |

Button Wiring diagram (Dr. Sudhu):

2 options, no wrong answer!



Chemistry Notes Pseudo-Code:

© MaryAnn Robak – Chem 3B Sp24 – Lecture Notes E – Multistep Synthesis with Carboxylic Acid Derivatives – page 21

```
//Define the first LED
int led = 2;

//Define button pin
const int ButtonPin = 13;

byte ButtonState = LOW;

//Define variable "Increasing" as true as a way to exit loop
int Increasing = 1;

void setup() {
  // initialize all pins as an output
  for (int led=2; led<6; led++){
    pinMode(led, OUTPUT);
  }

  // initialize button as an input
  pinMode(ButtonPin, INPUT);

  //all pins start off
  for (int led=2; led<6; led++){
    digitalWrite(led, LOW);
  }
}

void loop() {

  //defines variable "ButtonState" to read from pin 13
  byte ButtonState = digitalRead(ButtonPin);

  //current led starts on
  digitalWrite(led, HIGH);

  if (ButtonState != HIGH){ //if the button is not pressed
    digitalWrite(led, LOW); //turn off the current light
    if (Increasing == true){ //if the sequence is increasing
      led = led + 1; //move to the next light
      digitalWrite(led, HIGH); //turn on the new current light
      delay(100);
    }
    if (led == 5){ //if the last led in the row is the current light
      Increasing = false; //make the variable "Increasing" false
    }
  }
  else{ //If the variable "Increasing" is false
    led = led - 1; //move to the previous light
    digitalWrite(led, HIGH); //turn on the new current light
    delay(100);
    if (led == 1){ //if the first led in the row is the current light
      Increasing = true; //make the variable "Increasing" true
    }
  }
  else{ //if the button is pressed
    digitalWrite(led, HIGH); //stop at the current lit led
  }
}
```

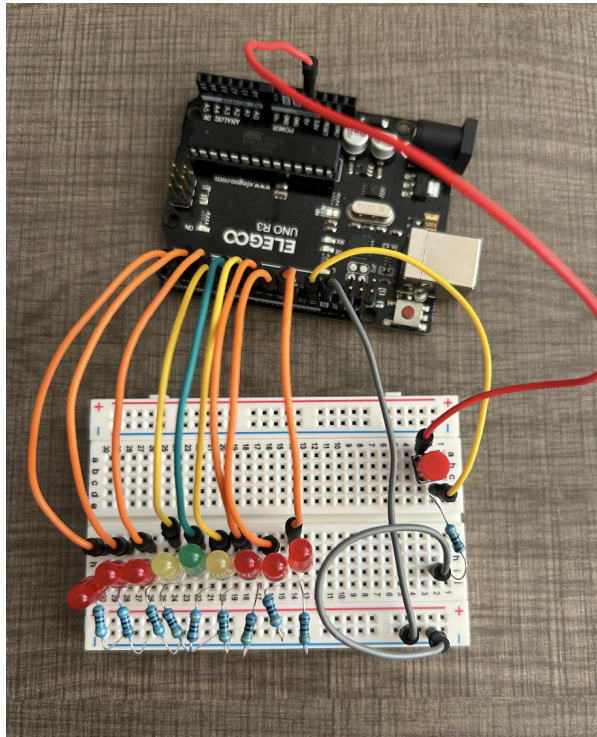
PART 2: Button

[Expressive Light Project: Button](#)[Stop the LED: Version 2 Code](#)

The second hurdle I faced was incorporating the button that would allow the light that was on at the time of the button press to remain pressed as the button was being held down. The first part of this was trying to figure out how to wire the button on the breadboard, for which I referenced an image included in Dr. Sudhu's slides (also pictured above). Once I felt pretty confident that the button was wired correctly, it was time to move on to coding this part of my project.

This step is where I felt quite a lot of frustration, mainly because I couldn't find a way to get my code to exit out of the for loop that I had going on. I spent a lot of time trying to figure out a way to do this, but ultimately wasn't able to. After spending about 30 minutes away from the project, while I was in my chemistry lecture, it suddenly came to me and I scribbled my "pseudo-code" (as Shm named it) on my chemistry lecture notes (image included above). Once I got back to my home later that day, I was

able to translate that into arduino code and it worked on the first try! I feel like I learned a lot from this process, particularly that sometimes it is much better to take a step back and restructure code approaching it from a different angle, than to try to work with what you have.



PART 3: Building it Out:

[Stop the LED: Version 3 Code](#)

Once I was able to get the lighting function and the button functions working in conjunction with each other, I had essentially made a mini working model of my game. The last thing to do was to build out a lot of the physical aspects to it, including adding more LEDs (and thus more resistors), organizing the colors to signal a “point system”, and organizing the wiring to make it much cleaner and easier to follow. In addition to this, I had to adjust a couple of the variables in my arduino code to have it fit with the new number of LEDs.

Conclusion / Reflection

I hope that at some point I will be able to further build out this game in order to make it a bit more aesthetically pleasing and user friendly. This would include perhaps building a little frame for the lights (perhaps with indicators that would note how much each light was worth), as well as using the serial monitor to allow for instructions and point tallies to be kept and displayed (took inspiration from Jashan's "Whack a Mole" project). Additionally, if this were to actually be used in an arcade, I would likely try to incorporate larger LEDs (which would probably require different resistors and power sources), as well as more types of LEDs. One great suggestion I received from a classmate was to perhaps adjust the resistors to have the red LEDs much dimmer and the green LED much bright so that you would also be able to see point values in brightness in addition to color as well. This might also help with people who might be color blind and allow them to use the brightness (rather than color) to help them indicate how close they are to the highest point value.

I also learned a lot of important lessons throughout working on this project, which I thought would be fitting to share here. (1) Like with writing, it might seem like it is unnecessary to plan and draw everything out before executing, but even in a project as small as this, writing things in pseudo code and drawing out how I wanted the project to look ultimately made things easier when it came time to execute. (2) When in a rut, it is important to take breaks and to try to step away from the project. There may be some inspiration that you can take from taking a break and giving yourself time might help bring some new and sudden ideas, saving a lot of time and suffering. (3) If in a huge rut, it can sometimes be a time saver to scrap your original code (of course, making sure that you save it somewhere just in case you need to come back to it) and to try to approach the problem from a completely different angle.

I am honestly very satisfied with my piece. Coming into this project, I had never had any wiring experience before, although I did have a bit of coding experience. Despite having this experience, I wanted to know how I could translate the code into something a bit more tangible and physical, so I am very proud of myself for learning to do that in such a short amount of time.

Links

Code	< Stop the LED FINAL.ino >
------	--

Final video	< Stop the LED Demo >
-------------	---------------------------------------

Sources

Cyclone Inspiration Photo	https://www.reddit.com/r/nostalgia/comments/6gpc97/the_cyclone_arcade_game/
Cyclone Inspiration Demo	https://www.youtube.com/watch?v=9-XAB7g7R84
Arduino LED Chaser Inspiration (both wiring & code)	https://www.youtube.com/watch?v=x4eh2gBwZSA https://youtu.be/e1FVSpkw6q4?si=LJ4iVUTR0HcWj3bi https://youtu.be/SJnW7U9kr_k?si=tWSuiDZXU-vbDDyh
Arduino Code Reference	https://www.arduino.cc/reference/en/
Dr. Sudhu's class 3 google slides presentation (used primarily for button wiring diagram)	https://docs.google.com/presentation/d/16h6vH_EWygwPfzliXOv4LFSDa26XWQzTSiNZMGQpqeE/edit#slide=id.g26704e3c49d_0_241
Dr. Sudhu's Arduino Tutorial	https://github.com/loopstick/ArduinoTutorial