University of Colorado Boulder
ECEE Department

ECEN 2270 - Electronics Design Lab - Spring 2024
Location: Engineering Center, ECEE 281, T,TH, 3:30 - 5:20 PM
Instructor: Steven Dunbar
Lab Title: Lab 5: Final Project (Courteous Driver)
Date of Experiment: May 2nd, 2024
Name: Connor Sorrell

Introduction:

In this lab, the final project, we focused on advancing the practical application of our electronic car by adding RGB lights to serve as headlights, taillights, brake lights, and turn signals. The chassis of the car was extended to enclose the breadboard. We did this using a variety of methods, including tape, cardboard, construction paper, and plastic. Utilizing Arduino as our controller, we intricately programmed the RGB lights through numerous tests to ensure seamless operation. The implementation involved using Arduino interrupts to detect the cars' movements, in which we could then dynamically add the light changes, mimicking real-world courteous driving. This final lab not only aimed to enhance the functionality and aesthetics of the car, but also served as a capstone project that took our cumulative engineering skills and challenged our creativity, in the end, resulting in a final car we are all proud of. The extensive application of circuitry and Arduino, as well as rigorous testing and problem-solving, have honed our ability to design, execute, and troubleshoot complex systems. Altogether, the culmination of this lab has significantly propelled and prepared us to be better, more adept engineers, capable of handling and innovating more complex tasks in the future.

Objectives:

Throughout the lab, there are some key objectives that are accomplished:

- The 4B7 Advanced Position Control Test is successfully completed across all four cars. All cars drive functionality and work as intended; drive in a straight line for a set distance and turn as expected using encoder pulses as an interrupt.
- New RGB lights are surface mount soldered to header pins using the proper solder technique.
- New RGB lights are tested and confirmed to work with Arduino.
- The new RGB lights are mounted to the newly built car.
- Using interrupts, Arduino code is programmed to develop fully functional, courteous driving car lights.
- Comprehensive debugging and refinement are done to ensure the light sequences are synchronized and performed as designed.
- Documentation of the coding and other modifications is completed, providing a record of the final project process.
- In the end, each car looks sleek and all lights work properly; including tail lights, headlights, brake lights, and turn signals.

Section 5A: Implementation

5.A.1: Preparing the lights

The lights we used were DIYmall 8 RGB LED Stick, 8 X WS2812 5050 RGB LED with Integrated Drivers. We chose these lights because they are individually addressable, already set up to work with an Arduino and look relatively simple to use.

Once we had the lights, we soldered 4 header pins to the surface pad of each side, the 4 pins being GND, 5V, DI, and GND. The soldering was relatively simple, however it was different from most of the soldering we have done in the past. In most applications, we have all only done through-hole soldering, so soldering to a flat surface had a small learning curve, but ended up being just as simple as we thought.

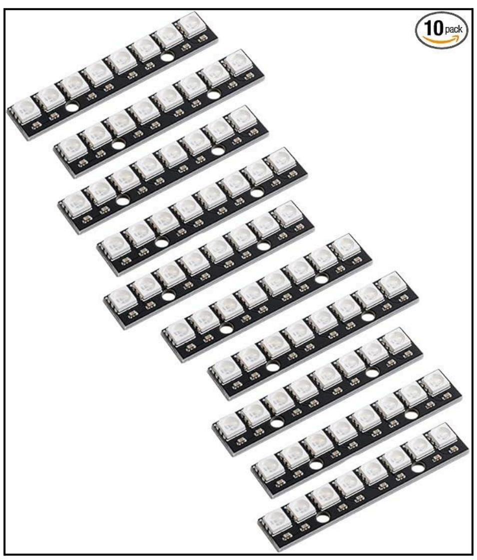


Figure shows one of the LED strips with the right side soldered and the left side unsoldered.

5.A.2: Testing the Lights

To test the lights, we first plugged them into the Arduino. We confirmed that the lights could work consecutively if the DI (Data Input) and DO (Data Output) pins were connected from one light to another. From there, we created light strands by connecting the pins of four consecutive light strips together, making every light controllable with a single Arduino digital pin. Furthermore, there were two ground connections available on the LEDs; however, only one was necessary to properly power and daisy-chain the rest of the lights together. This reduced the amount of wiring protruding from our design and saved space on the breadboard.

To ensure proper functionality, we implemented a series of tests. We started by running simple scripts to light up each LED individually, verifying that each could display the full RGB spectrum without faults. Next, we tested the responsiveness of the LEDs to dynamic changes in the code, such as varying brightness and color transitions, to simulate real-world traffic scenarios like braking or turning signals.

An additional focus was placed on the integration of these light systems with the encoder pulse interrupt, allowing the lights to react to the cars movements, such as detecting the car coming to a halt, or the car initiating a turn. An important detail for us was to make sure that the brake lights could be triggered automatically.

Lastly, peer reviews were conducted within our team. Each member developed different code and different cars to provide feedback on the functionality from various different methods. In the end, we ensured that the implementation was effective, although not consistent across all cars.

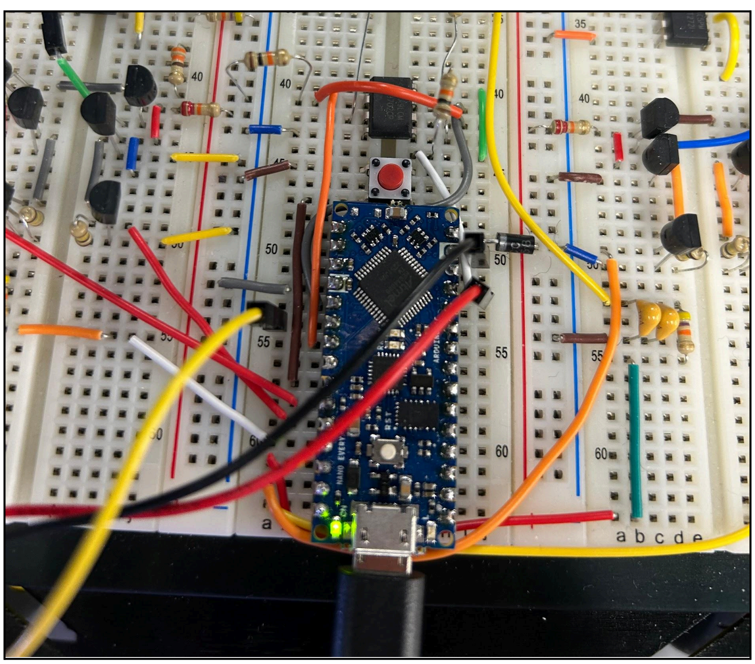


Figure shows the lights connected to Arduino pin D3

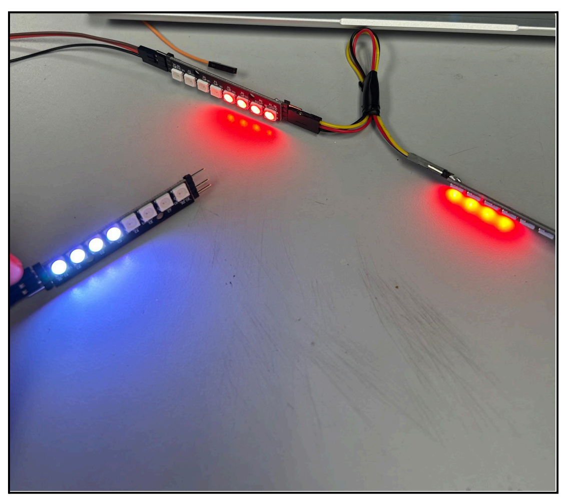


Figure shows the daisy chaining of the lights

5.A.3: Car Design and Light Attachment

It was decided across the group that each person would design a casing for their RC car inspired by military and police vehicles. To achieve the design, precise measurements were necessary, as well as considerations for practical complications such as blocking access to buttons and switches needed to turn on and start the car. Furthermore, cable management became more tedious and difficult to manage around the body or underneath the encasing for a more sleek appearance. The material for the car was chosen to be either cardboard or plastic because both materials are affordable and highly customizable.

1. We had to relocate the power button to ensure it remained accessible even after the casing was built. This required careful cutting and positioning within the design to allow for easy reach without compromising the structural integrity of the casing.
2. Keeping the Arduino freely accessible was crucial; we all left a hatch open on the front of the casing that allowed for the USB to still be plugged in and uploading of code without needing to disassemble the car.
3. Cable management was improved by bundling wires together and using clips and straps to secure them within designated channels inside the casing, which helped maintain a tidy appearance and prevent interference with the vehicle's moving parts.
4. The lights were then integrated into the casing, emulating two headlights and two tail lights. We carefully routed and concealed the wires of the lights to enhance the aesthetic appeal and prevent any loose ends that could be caught by the tires during the movement of the car.
5. Overall, while the cars may not look perfect, they are significantly more aesthetically pleasing than before. The effort to customize and refine the design of the casings added a professional touch that more closely mimicked a real vehicle, achieving a balance between functionality and visual appeal.

5.A.4: Coding the lights

First, we needed to download the FastLED library, which provided a selection of colors for the lights and several pre-built functions that significantly simplified our coding efforts. Our goal was to design Arduino code to turn on, customize, and successfully use the LED lights to simulate blinkers, headlights, and brake lights as our car navigates the 4.B.7 advanced position course. To achieve this, we used the encoder count and pulses to determine when to enable the brake lights and blinkers. Once the pulses reached a predetermined value, the brake lights and blinkers would activate automatically.

To further enhance the looks and functionality of our car's lighting system, we incorporated conditions to adjust the brightness and flashing patterns. For example, on Connor's car, the brake lights mimic sequential tail lights, where the blinker runs across the tail light. Additionally, we programmed the headlights and tail lights to always be on, similar to running or daytime lights. All these functionalities were tested under various scenarios on the track to ensure they performed reliably and enhanced the overall “coolness” of the car.

Shown below are examples of code we used for the lighting effects.

```
enc_count = 0;
  target = 2550; // 2500 gets me 2 ft exact
                  // 1250 pulses per foot
  digitalWrite(pinLeftForward, HIGH);
  digitalWrite(pinRightForward, HIGH);
  do {
   while (enc_count > 1250 && enc_count < 2550) //Using encoder pulses
    {
      Serial.println(enc_count);
      blinkersright();
      breaklights();
    }
  } while(enc_count <= target);

  breaklightsoff();
  digitalWrite(pinLeftForward, LOW);
  digitalWrite(pinRightForward, LOW);
```

<u>Brake Lights:</u>

```
void breaklights()
{
  leds[12] = CRGB::Red;
  leds[13] = CRGB::Red;
  leds[14] = CRGB::Red;
  leds[15] = CRGB::Red;

  leds[16] = CRGB::Red;
  leds[17] = CRGB::Red;
  leds[18] = CRGB::Red;
  leds[19] = CRGB::Red;
  FastLED.show();
  return;
}
```

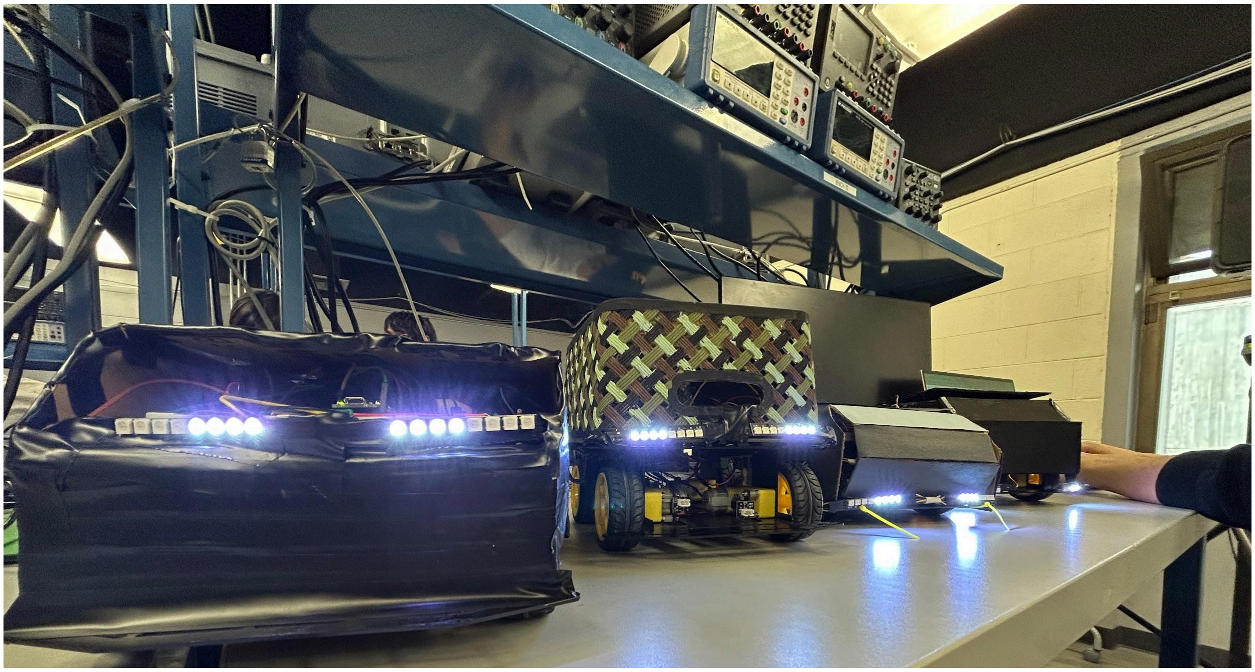
Blinkers:

```
void blinkersright()
{
  // Turn the LED on, then pause
  leds[0] = CRGB::Orange;
  leds[1] = CRGB::Orange;
  leds[2] = CRGB::Orange;
  leds[3] = CRGB::Orange;

  leds[8] = CRGB::Orange;
  leds[9] = CRGB::Orange;
  leds[10] = CRGB::Orange;
  leds[11] = CRGB::Orange;

  FastLED.show();
  delay(150);
  // Now turn the LED off, then pause
  leds[0] = CRGB::Black;
  leds[1] = CRGB::Black;
  leds[2] = CRGB::Black;
  leds[3] = CRGB::Black;

  leds[8] = CRGB::Black;
  leds[9] = CRGB::Black;
  leds[10] = CRGB::Black;
  leds[11] = CRGB::Black;
  FastLED.show();
  delay(150);
  return;
}
```

By the end, we were able to simplify our code, turning each individual action into its own function. This made it much easier to read, test, and fix the code. The code below shows the simplified code for completing the 4B7 Advanced Position Control Test with courteous lighting.

```
void loop()
{
 do {} while (digitalRead(pinON) == HIGH);
 Headlights();
 Taillights();

 delay(1000); // straight
 RunStraight(1000);
 BrakeLights();
 LeftTurnSignal();
 delay(1000); // left turn, go straight
 TurnLeft(650);
 RunStraight(2000);
 BrakeLights();
 RightTurnSignal();
 delay(1000); // right turn, go straight
 TurnRight(600);
 RunStraight(1000);
 BrakeLights();
 RightTurnSignal();
 delay(1000); // right turn, go straight
 TurnRight(600);
 RunStraight(2000);
 BrakeLights();
 RightTurnSignal();
 delay(1000); // right turn, go straight 2ft
 TurnRight(600);
 RunStraight(1000);
 BrakeLights();
 RightTurnSignal();
 delay(1000); // right turn, go straight 4 ft
 TurnRight(700);
 RunStraight(2000);
 etc……..
}
```

5.A.6: Finished Design



Section 5B: Wrapping it all up

Successes:

- Successful completion of the 4B7 Advanced Position Control Test across all four cars, demonstrating their ability to drive in a straight line for a set distance and turn as expected using encoder pulses as an interrupt.
- Surface mount soldering of RGB lights to header pins, showcasing proficiency in soldering techniques.
- Comprehensive testing and confirmation of RGB lights functionality with Arduino, making sure the operations are working as expected before integration into the cars.
- Implementation of Arduino interrupts to detect car movements and dynamically control light changes, mimicking a real-world driving scenario.
- Design and construction of casings for each car completed and look aesthetically pleasing.
- Integration of lights into the casings, providing aesthetics while maintaining functionality and accessibility of the components.
- Development of Arduino code to control and customize LED lights for simulated blinkers, headlights, and brake lights, enhancing the cars visual appeal and functionality during the driving test.

Failures:

- Inconsistencies in RGB lights; random blinks can occur, the arduino code has some faults.
- Simplification of design choices due to time constraints, such as the removal of the Bluetooth controller feature. In addition, simple materials like plastic and cardboard were used for the casing rather than a sturdier option such as wood. Time constraints limited the scope of the project.
- Challenges in managing cable routing and structural integrity within the casings, leading to minor design compromises and aesthetic imperfections, such as having the arduino button in a slightly awkward position.
- Limited RGB creativity; rainbow effects and moving effects were a trouble to implement because of our inexperience with code.

5.B.2: Takeaways / Conclusions

If we had more time:

Over the course of Lab 5, design choices and implementation methods were made simpler for the sake of time management. If we had more time, other design choices would have been made and pursued. First of all, given the time crunch, we decided to simplify our original plan by removing the Bluetooth aspect and focusing on the courteous LED robot. We feared that pursuing both design choices would be too ambitious for time's sake. Given the focus on just a courteous LED robot, we were able to fully develop a courteous robot. Provided more time, the first design choice we would reimplement would be the Bluetooth controller. We wanted to make a Bluetooth controller that would control the movement of the vehicle as well as the headlights and turn signals. The next design choice we would change would be in the implementation of the brake lights. As discussed in the Implementation section, our brake lights were lit dependent on pulse counts. If we had more time, we wanted to implement brake lights that were dependent on the integration signal received in the feedback loop. Remember that when the reference voltage is too high, the feedback loop causes deceleration. Programming this signal to the Arduino, we would've implemented brake lights that lit as soon as deceleration began to occur. This would greatly decrease the error in brake light on and off time.

Takeaways / What Did We Learn

We learned a multitude of information throughout the process of creating our Lab 5 project. The majority of the project was software-related, specifically Arduino, yet some hardware skills were also developed. The hardware skill we mainly developed was flat-pad soldering. The LED boards we used came with flat-pad bases. Using much care, we soldered header pins to these flat-pad boards. This skill will prove useful in the future, as many boards come in a flat-pad manner. The main aspect of this project was software-related, utilizing Arduino code to run the LED operations. Because of this, we greatly enhanced our software proficiency, specifically Arduino programming for controlling LED lights and implementing interrupts for real-time responses. This is undoubtedly a very important skill and we are happy that we have taken our coding effort to a higher level than what we were taught in the scope of the class. During our final project, we also recognized the value of collaboration and peer learning, as we refined our designs and troubleshooted together, helping each other understand certain topics or even fix components of the car.

Advice for future students:

For future students, it is crucial that they choose an attainable design given such a limited amount of time to develop and create the final project. With only two weeks it can be difficult to order all the necessary parts and ensure they'd work for their intended purpose. Implementing this design takes time and working with code that can be unfamiliar will certainly extend the hours necessary to complete the project. The most important thing future students should be aware of is the time constraints in place to avoid unnecessary stress or ambitious goals that are not feasible. It's also important to communicate and collaborate with lab mates in an effective manner, especially when all pursuing a similar final project. This helped us greatly as we all tried to grasp certain concepts that were not taught in class. Finally, an extremely important aspect of this lab is to allocate sufficient time and energy for testing and refinement. When using creativity in order to create a practical design, many problems and roadblocks are realized and testing and fixing errors to ensure functionality and consistency can take much longer than anticipated.

Broad Accomplishments:

Overall, this lab in its entirety served as a massive culmination of all of our engineering skills. It challenged us to apply theoretical concepts as well as hands-on laboratory work in order to problem solve in a practical, real-world context. Through months of testing, validating, modeling, circuity, design implementation, and arduino programming, we developed countless valuable skills which have propelled our confidence in tackling more complex projects in the future. In addition, the nature of the lab being so collaborative taught us many intangible skills relating to teamwork, communication and flexibility. As a whole, we all feel confident that this class as a whole has greatly prepared us for our future engineering endeavors.

Final Product:

Connor Sorrell:

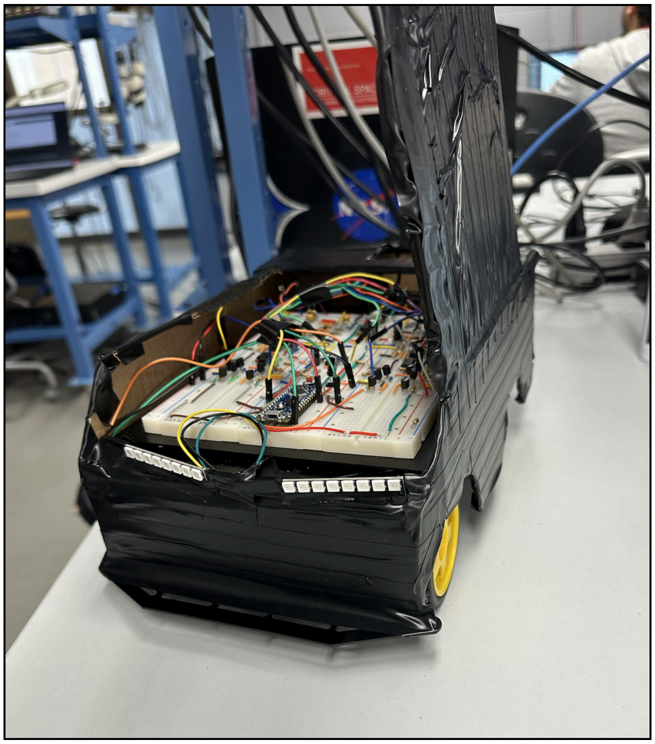


Figure shows all 4 cars fully built and functional.