Introduction:

The objective of this project was to construct a small vehicle that could follow both circular and straight line paths made of black electrical tape. We broke this larger objective into smaller steps, the first of which was setting up the sensors. Knowing that electrical tape reflects infrared light, we created infrared LED and infrared sensor pairs to take readings of the light reflected from below the car. Large sensor readings indicated that a given sensor was directly above the path, while lower readings indicated that it was over plain white paper. By placing these sensor pairs on the left, middle, and right of the center of the vehicle, we were able to determine where the car was relative to the path.

Next we worked on the code that would control the car. We chose to pursue the LMP design, which used an Arduino Nano programmed with a proportional-integral-derivative (PID) controller that was connected to three infrared sensors and one hall effect sensor to detect magnets under the tape. We used the concept of minimizing error, defined as distance from the path, to design the controller. We learned about how PID works with the change in error to create smoother driving on behalf of the car. We also installed motors to drive the wheels and wrote code such that inputs from the sensors would trigger the correct responses in the motors, which were controlled with an H-bridge. We programmed the Arduino to send alternating signals to the H bridge in order to control the speed and direction of rotation of each motor.[3] This allowed the car to make turns. After confirming that we could properly read from our sensors and control the car, we developed our PID code.This project tested our knowledge of designing circuits with many sensors and our ability to program microcontrollers to make decisions based on the sensor readings.

Testing Methodology:

In order for our PID code to function correctly, we needed to test how the phototransistors in the infrared sensors would respond when the vehicle was on and off of the track. Below we outline our methods of testing the phototransistor responses.

How We Designed the Test

Each infrared sensor was already properly mounted on the vehicle and connected to the Arduino. We needed to find values outputted by each phototransistor for various positions to the left and right of the path. To mimic the path the vehicle would actually be running on, we simply used a piece of paper with a small strip of electrical tape on it. We decided to take measurements at two millimeter increments up to 20 mm to the left and right of the center of the tape. This is what was recommended by the professor and we thought any measurements more frequent than every two mm would not contribute significantly to the general trend we sought to quantify. We needed to record our data, so Arduino code was written to output the values read by each sensor to our computer screen. For the processing of the data we decided to then plot the collected data to associate each sensor output with the corresponding distance of the vehicle’s center from the center of the electrical tape. This association would be used to determine the ranges of sensor values that would correspond to the correct positional error of the vehicle, which is necessary for the PID controlling of the vehicle.

How We Conducted the Test

First, we marked 2mm increments to the left and right of the paper with the strip of electrical tape. These marks represented each position that we would collect data for. We also mounted shrouds over the phototransistors before collecting data, and marked the center of the vehicle with a vertical mark in the center-front of the shroud for the middle phototransistor. Placing the vehicle on top of the paper and moving it so its center aligned with each point, we took data at 21 points, spanning 20 mm on either side of the tape. The design of the shrouds and the spacing of the infrared sensors are the only facets of the vehicle that changed from the testing. The left and right phototransistors, with the infrared LEDs behind them, were initially too close to the middle of the vehicle. They were both moved with their respective infrared LEDs behind them, about one millimeter further away from the middle sensor. The length of the shrouds had to be changed as well. The shrouds were initially too long, so the phototransistors could only detect the tape if they were directly over it. The shrouds were slightly shortened, and this solved the problem.

How We Analyzed the Test Data

The outputs of the three mounted infrared sensors were recorded as the vehicle was moved to positions at varying distances from the center of electrical tape. These output values were recorded in excel with the corresponding distance in millimeters, in order to determine the trend between each sensor output and the position of the vehicle. The data collected is shown below.

Distance of Vehicle Center from Tape Center (mm)	**Left Sensor Output**	**Middle Sensor Output**	**Right Sensor Output**
-20	975	975	472
-18	974	975	378
-16	974	975	338
-14	974	971	309
-12	974	970	332
-10	973	796	344
-8	973	565	729
-6	973	514	973

-4	973	468	977
-2	973	439	977
0	976	505	979
2	973	523	979
4	956	553	980
6	911	631	980
8	645	946	980
10	373	970	980
12	317	973	980
14	287	975	979
16	284	976	979
18	287	975	980
20	299	976	980

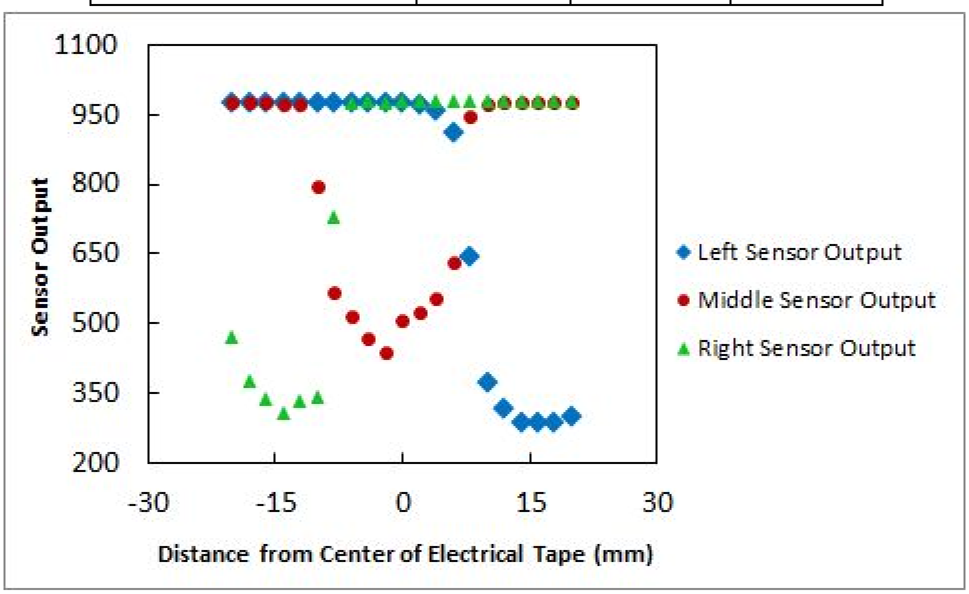


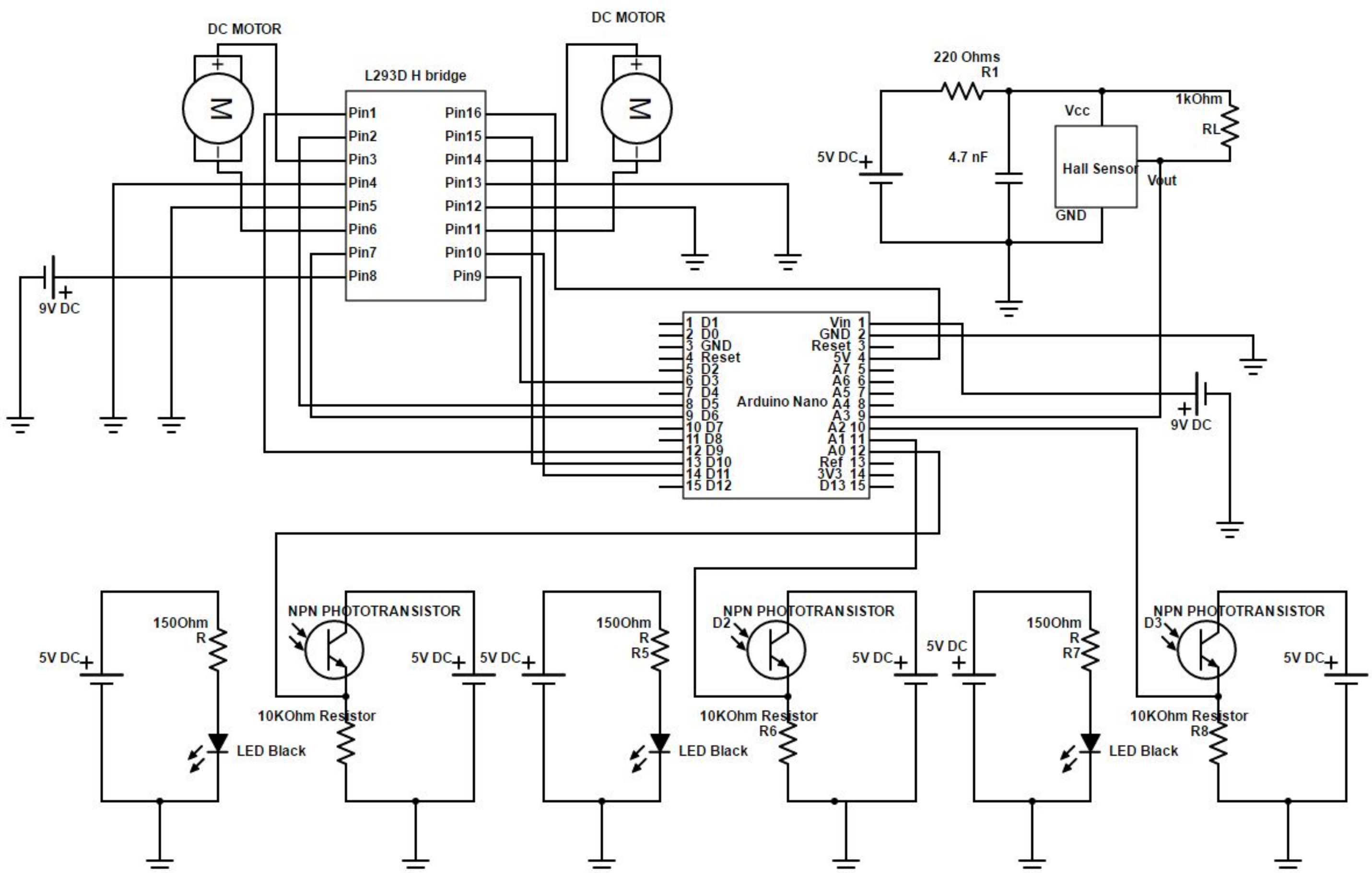
figure title will be added in official doc

No equations were applied to this data, due to the unideal responses from the phototransistors in the infrared sensors. The data did not fit any curve, or mathematical model that could have been used to calculate the positional error of the vehicle.

We can see three parabolic shapes in the graph, representing the

How We Interpreted the Data

The left and middle sensors behaved as expected, but the output from the right sensor increased when the car was more than 15 mm to the left of electrical tape. The increase in output from the middle sensor when moving right from zero to two millimeters away from the tape was much steeper than the increase when moving left. This asymmetry and the unchanging outputs from the left and right sensors made it impossible to calculate accurate and precise positional error when the vehicle was two or less millimeters away from the center of the tape. The precise calculation of positional error when the vehicle was left of the tape was also impossible, due to the increase in the output from the right sensor as it went further than 15 mm away.



Results and Discussion:

Only one test required data to be collected, and this was the test of the infrared sensors that is described in the above section. The data collected from each sensor is explicitly shown in the table and graphically presented in the plot from the testing methodology section.

Due to the outputs from the sensors not being ideal, using a curve with mathematical equations to determine positional error of the vehicle would not have been effective as the actual readings from the sensors would not fit this model. The best way to determine the vehicle's positional error was to use ranges of output for the left and right sensors that would correspond to the error values zero through four, with negative error denoting the vehicle is to the left of the tape. The ranges set are seen in the arduino code, and sum of the error produced from each sensor would be used to calculate the final positional error. Individual if-else-ladders for the left and right sensor were used instead of one longer if-else-ladder that would evaluate the output from both sensors. This choice was made to reduce the number of comparisons the arduino would have to make. The middle sensor would just be used for determining whether or not the vehicle was on the track, and this value is used in a mechanism that helps the vehicle find the path if it moves off of it.

The only other quantitative test involved the hall effect sensor. We placed a magnet at varying distances from the sensor, taking note of the values output by the sensor, and the maximum distance at which the sensor would respond to the magnet. We also looked for a threshold value at which we could be sure that a magnet was close to the car. After conducting the test, we found that the sensing distance was about 1.5 centimeters, and within this range the sensor would output a value of at least 500. The hall effect sensor was then mounted to the bottom of our vehicle so it would be about one centimeter from the ground, and the Arduino was programmed to light its integrated LED while the sensor output is a value greater than or equal to 500.

Conclusions and Future Work:

Our final design met our initial goals very well. The car was able to follow both the straight and circular paths, and was able to sense all five magnets on the straight path. We learned how to interpret multiple sensor readings to program a system to make decisions. We also learned how to set up these circuits with an arduino, which we programmed with their modified version of C. We also learned how H-Bridges allow for selective control over outputs.

Given more time, we would try to increase the speed of the car. We could not simply change the motor output to make it go faster as that would affect our sensor system. We would have to reposition the sensors such that faster changes in direction do not take the car off the track and also add to the code to make it interpret readings faster. We would also work on starting the car outside of the track and having it acquire the path. We had success with this on the straight path and the circular path going clockwise, but had trouble getting it to work on the circular path going counterclockwise. In order to do this we would need to adjust our PID controller code.

Illustration credit:

All illustrations were developed by the authors of this report.

Reference:

1. Arduino.Nano.Schematic [pdf] [Online]. Available from: https://www.arduino.cc/en/uploads/Main/ArduinoNano30Schematic.pdf

2. DigiKey. SchemeIt. (Online) [Software] Digi-Key Electronics. Available from: https://www.digikey.com/schemeit/project. 2017

3. Texas Instrument (JANUARY 2016). H-Bridge Data Sheet [pdf] [Online]. Retrieved June 17, 2017, Available from http://www.ti.com/lit/ds/symlink/l293.pdf

Computer program, software or code

- **Author** (use the corporate author or research group if no individual author or editor is named)
- **Title of program** (this should be in italics)
- **(Version number)**
- **[Format type]** (computer program, software or code)
- **Place of publication** (if available)
- **Name of publisher/distributor** (if available)
- **Available from: URL** (if online)
- **Year of publication**
- TechSmith. Snagit. (Version 9.1) [Software] TechSmith Corporation. Available from: http://www.techsmith.com/screen-capture.asp. 2008.

Digi-Key Electronics. SchemeIt. [Software] Digi-Key Electronics. Available from:

Our test for the Hall sensor was very similar. We planned to place a magnet at varying distances from the sensor and record the values read by the sensor. We looked for a threshold value at which we could be sure that a magnet was close to the car.

After moving the magnet to various spots, we found that the sensing distance was about 1.5 centimeters. This corresponded to a sensor reading of about 500, so we set that as the value which indicated that a magnet was close by.

For the Hall sensor, we decided to light up and LED when the sensor detected a magnet. To test the hall effect sensor, we connect the circuit on the breadboard with a 9 volt output and put the magnet close to the sensor. The value shown on the arduino readings drops when the sensor gets close. A threshold value was determined to light up the LED if a magnet is detected.