A Mobile, Path-Following Robot

Shawye Ho
Byunghun Lee

<u>Introduction</u>

The goal of this project was to build a mobile robot capable of navigating a simple path marked by dark tape on a while piece of paper. We divided this project into several components, using a provided Arduino Nano to control and monitor them. One component was the motors would need to be added to allow the mobile robot to move. Another component was the three sets of infrared light-emitting diodes (LEDs) and infrared transistors which were attached to the bottom of the robot; this gave us information on whether a certain portion of the robot was above a light or dark area. This information allowed us to have the robot follow the dark-tape marked path. We also added a red, green, and blue-colored LED to show a viewer the bot's current action.

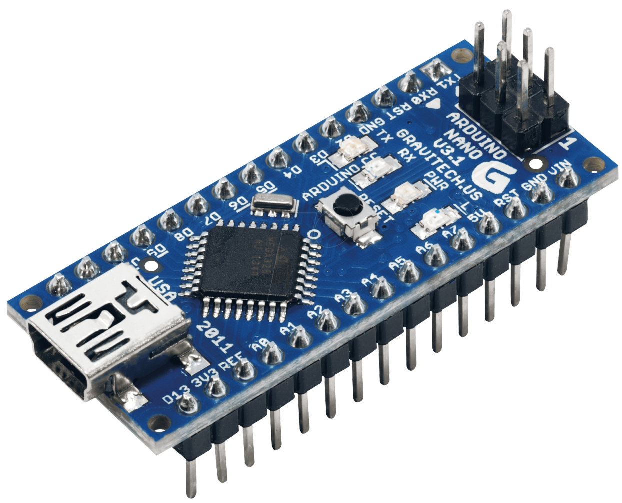

Figure 1. The Arduino Nano
Image taken by reichelt elektronik GmbH & Co. KG, from
https://cdn-reichelt.de/bilder/web/xxl_ws/A300/ARDUINO_NANO_03.png

The Arduino Nano (Figure 1) that was provided is a small board that can be attached to a micro-USB cable, allowing code to be uploaded and run. The Arduino was attached to the breadboard, and hooked up a 9V battery through its voltage-input (VIN) pin. All components were connected to the Arduino Nano through an appropriate pin, as shown on the Arduino Nano Pinout Datasheet[1], with digital pins capable of generating a pulse-width modulation (pwm) signal used for output, and analog pins used for input. The motors for the robot were attached to a 6V battery through a TIP120 NPN Darlington Transistor to amplify the current, by consulting the TIP120 NPN Darlington Transistor Reference Sheet[2] and using the circuit shown in Figure 2.

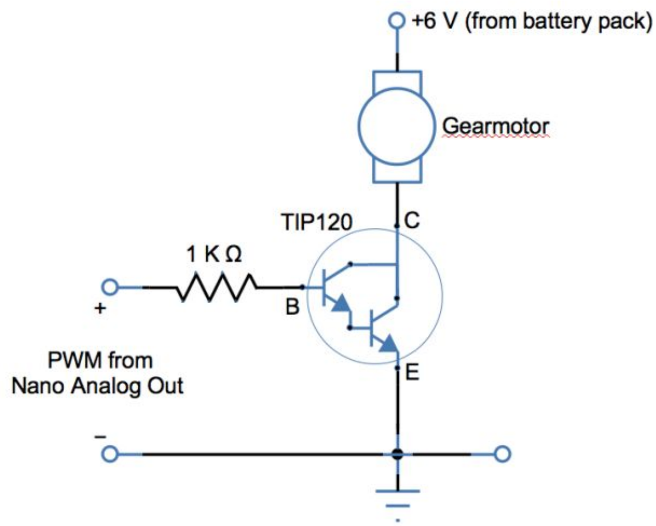

Figure 2. Circuit used for mobile robot motors
Image created by Briggs, Dennis, taken from EE3 Lecture *Project Prof Pres 17F*

Three parallel infrared LED-transistor circuits (each of these circuits had an infrared LED next to an infrared transistor, and they were wired in parallel circuits properly) were wired on the front underside of the mobile robot: one on the left of, one on the right, and one in the center, so sense the position of the dark path relative to the robot. These infrared transistors were wired in the form of an emitter-follower circuit (Figure 3).

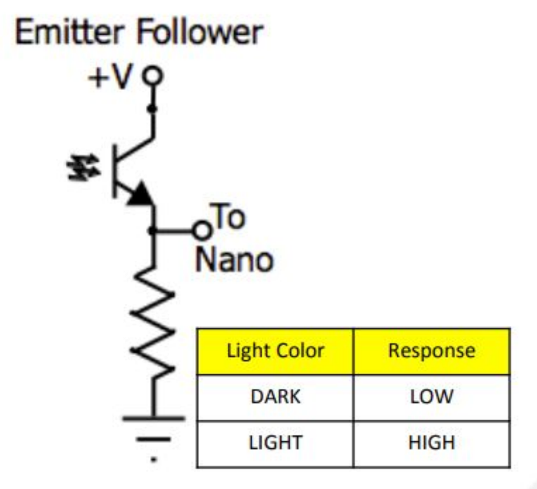

Figure 3. Emitter follower circuit used for infrared transistors
Image created by Briggs, Dennis, taken from EE3 Lecture *Project Prof Pres 17F*

As seen in Figure 3, this setup will allow current to flow to the Arduino Nano if infrared light was hitting the transistor. Thus, the transistors to have a low response when near the dark tape, where there was a low amount of infrared light reflected from the LED, and a high response when near white paper, where greater amounts of light were reflected from the LED. All LEDs, infrared and colored, were connected in a typical circuit that included a voltage source and dropping resistor, which helped dissipate excess voltage to the LED (Figure 4).

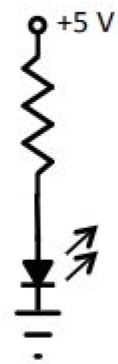

Figure 4. Typical LED circuit
Image created by Briggs, Dennis, taken from EE3 Lecture *Project Prof Pres 17F*

The resulting configuration was similar to that shown in Figure 5.

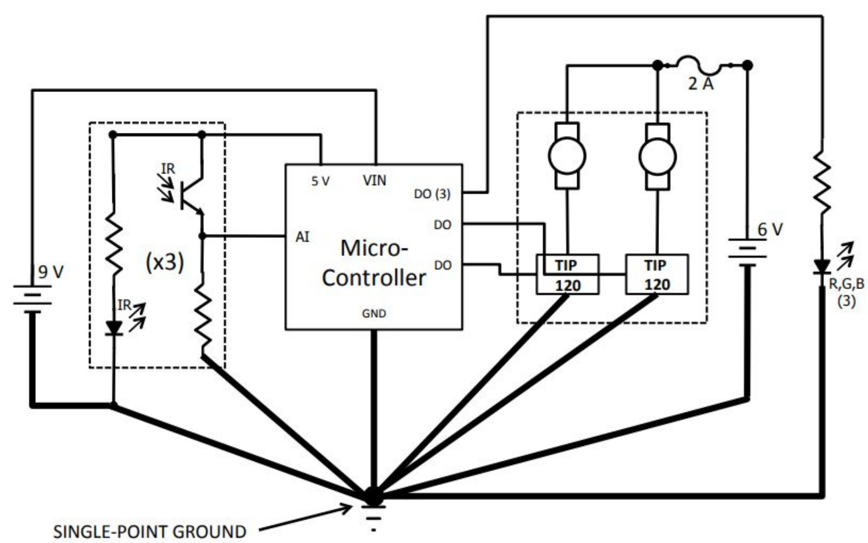

Figure 5. Complete block diagram of mobile robot circuits
Image created by Briggs, Dennis, taken from EE3 Lecture *Project Prof Pres 17F*

Testing Methodology

Testing the Path Sensing Subsystem

How We Designed the Test

The most difficult - and important - part of the path sensing subsystem was ensuring that each LED-transistor pair could correctly sense whether or not they were above the marked dark path or above the white paper. To test this, we obtained a piece of paper with a printed black line the width of the path. This was used to simulate the path. We would record the transistor response values with the black line portion of the paper, then the white portion of the paper. If the each set of LED and transistor were working properly, we would expect to see a significant response difference between the black line and white paper backgrounds. To do this, we wrote a program using the Arduino IDE software[3], with help from the function list[4], that would print out the response values, as demonstrated[5]. Then we could record the response of the transistor as we changed from the black line to white paper. We tested several different circuits, with differing resistors.

How We Conducted the Test

After uploading the appropriate program (coded with help from the function list[4]) to the Arduino Nano using the Arduino IDE software[3], we placed the black line beneath the LED and transistor set we were testing, and recorded the response values (Figure 6).

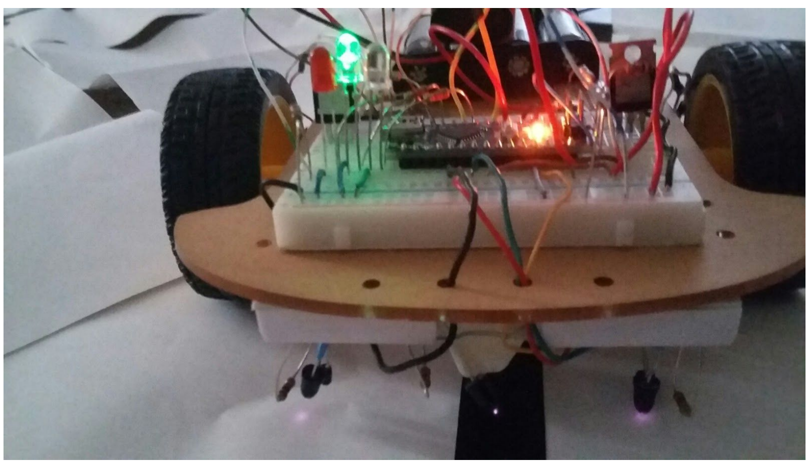

Figure 6. Testing the LED and transistor of the middle sensor. The black line is placed below the middle sensor to get readings for response values for when the black line (marked path) is below the sensor
Image taken by the author (Byunghun Lee)

Then, we moved the paper such that the LED and transistor were above white paper, and recorded the response values (Figure 7).

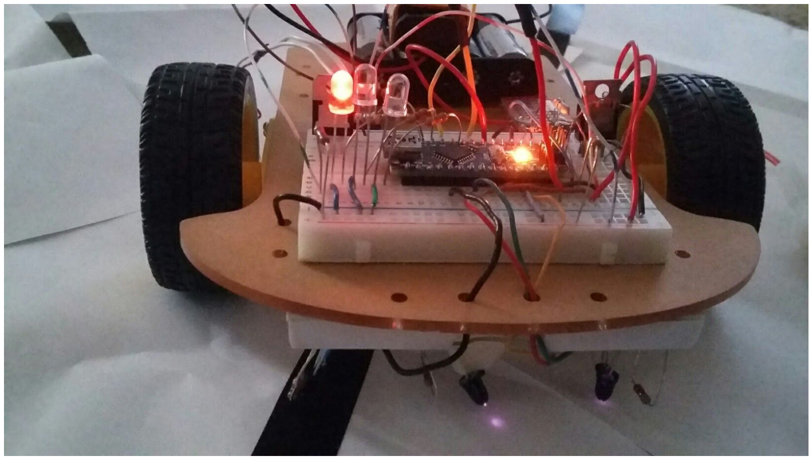

Figure 7. Testing the LED and transistor of the middle sensor. Instead of a black line, white paper is placed below the middle sensor to get readings for response values for when the white paper (off the marked path) is below the sensor
Image taken by the author (Byunghun Lee)

We repeated this procedure for each LED-transistor pair we tested, and for each time we changed the value of the resistor in the LED and transistor circuits. The data is shown below, in the <u>Results and Discussion: Test Discussion</u> section.

<u>How We Analyzed the Data</u>

Because of our method of printing out a string of response values (as demonstrated in [5]), after each result, we were left with a string of numbers. However, we knew which response values were associated with the black line, and which response values were associated with the white paper. This allowed us to consolidate the data into a table, with each entry in the table indicating the range of response values associated with either the black line or white paper. An example is shown below. In Figure 8, the response value data for one of the sample tests is shown. By taking the range of the response values of the transistor when above light white paper or the dark black line, a table can be created showing this data, shown in Figure 9.

12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 13 12 13 13 14 14 14 14 14 15 16 16
17 18 20 21 22 24 26 28 30 33 35 38 40 43 46 48 50 52 53 53 54 55 56 56 56 56 57 57 57 57 57 58 58 58 58 58 58
58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58
58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 59 59 59 58 59 59 59 58 59 59 59 58 59 59 58 59 58 58 58 58 58 58
58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58

Figure 8. Data for response values from the left transistor when 1000 ohm resistors were used for both the LED and transistor circuits. The first 70 numbers are the response values with the black line underneath the sensor, while the last 70 numbers are the response values with the white paper underneath the sensor. Thus, we can analyze the data and condense it into a simple table.

Light Color	Response Value
Light	58-59
Dark	12-14

Figure 9. Sample table using data from Figure 10, light and dark response values for the leftmost transistor, with 1000 ohm resistors used for both the LED and transistor circuits.

How We Interpreted the Data

What we want to measure is the difference between the light and dark response values; that is, light - dark. We certainly expect that this number would be greater than zero because the transistors are in an emitter follower setup, so the light response should be high, while the dark response should be low. If the difference between the light and dark response values is less than zero, it would indicate that there was an error in either our LED or transistor circuit, or both. Otherwise, any significant positive value is a success, as it would indicate that there is a difference between the light and dark response values. Optimally a larger value is better, since it will allow the system to be less sensitive to changes, since the difference in light and dark response values will be more clearly defined. We expect there to be more differentiation between response values if the transistor has a higher resistor, since having a higher resistor in our emitter follower transistor circuit would lead to a much larger response detected by the Arduino Nano. This is because a larger resistor in the emitter follower transistor circuit would cause more voltage to go the Arduino Nano, resulting in a larger response.

Results and Discussion

Test Discussion

The data for different tests is shown in Figures 9-14.

Light Color	Response Value
Light	41-43
Dark	19-21

Figure 10. Light and Dark response values for the middle transistor, with 1000 ohm resistors used for both the LED and transistor circuits.

Light Color	Response Value
Light	56-59
Dark	9-10

Figure 11. Light and Dark response values for the rightmost transistor, with 1000 ohm resistors used for both the LED and transistor circuits.

Light Color	Response Value
Light	900-989
Dark	198-201

Figure 12. Light and Dark response values for the leftmost transistor, with 1000 ohm resistors used for the LED and 47,000 ohm resistors used for the transistor in their respective circuits.

Light Color	Response Value
Light	317-321
Dark	159-161

Figure 13. Light and Dark response values for the middle transistor, with 1000 ohm resistors used for the LED and 47000 ohm resistors used for the transistor in their respective circuits.

Light Color	Response Value
Light	970-972
Dark	238-262

Figure 14. Light and Dark response values for the rightmost transistor, with 1000 ohm resistors used for the LED and 47000 ohm resistors used for the transistor in their respective circuits.

Since it is our goal to find the difference between the light and the dark response values (light-dark), we can combine this into another table showing the difference for different setups to help interpret the data. The maximum response difference will be the largest light response subtracting the smallest dark response; the minimum response difference will be the smallest light response subtracting the largest dark response. This is shown in Figure 15.

Setup (Sensor Position, LED Resistor, Transistor Resistor)	Response Difference (Light Response - Dark Response)
Left, 1000 Ohm, 1000 Ohm	44-47
Middle, 1000 Ohm, 1000 Ohm	20-24
Right, 1000 Ohm, 1000 Ohm	46-50
Left, 1000 Ohm, 47000 Ohm	699-791
Middle, 1000 Ohm, 47000 Ohm	156-162
Right, 1000 Ohm, 47000 Ohm	708-734

Figure 15. Response differences for different sensors with differing resistor for the infrared LED and infrared transistor circuits. The difference range was calculated to be such that $Response_{max} = Light_{max} - Dark_{min}$ and $Response_{min} = Light_{min} - Dark_{max}$

Looking at Figure 15, it is obvious that using a much larger resistor for the transistor circuit would be more optimal. For each sensor, using a large resistor for the transistor circuit results in a considerably larger response difference: for the leftmost resistor, this response difference goes from 44-47 to 699-791, an increase of over 500. For the middle resistor, this response difference goes from 20-24 to 156-162, a modest increase of over 100. For the rightmost resistor, this response difference goes from 46-50 to 708-734, an increase of over 500. As discussed above in <u>Testing Methodology: How we interpreted the data</u>, the best case is a larger response difference, since it will be easier to differentiate from the black line marking the path and the white paper. Thus, since there was a larger response with the 47000 ohm resistor in the transistor circuit, as expected, that circuit was used.

Race Day Results

The most unforgiving portion of the project was probably choosing the correct resistors for the LED and transistor circuits. We found (in the tests described above) that a low resistor for the LED and a large resistor for the transistor gave the most response difference, and this allowed our robot to more easily run the track and be less sensitive to changes, such as those that might come from hitting a bump in the paper. However, this decision came after several failures in the setup previously. The biggest problem with this is when a mistake is made in any other component of the robot, it is noticeable immediately: forgetting a dropping resistor will cause the LED to burnout; an incorrect circuit for the wheel motor or LED will cause them to fail to operate; an incorrect circuit for the wheel sensor will cause the sensor to display nonsense. However, choosing the wrong resistors for the infrared LED and transistors does not immediately raise any red flags. Only after failing to achieve consistent results does the mistake become apparent; even then, the mistake is not obvious. Thus, before beginning to instruct the robot to navigate the path by writing code, the correct resistors must be carefully chosen, or else the project will become much more difficult, and the mistake will be hard to find.

On Race day, we suffered several setback when the code that had previously been able to complete the course failed, and the motors were not moving fast enough. Furthermore, the sensors had become decalibrated. We were able to overcome this by first calibrating the sensors. Then, we recalibrated the motors with a significantly higher pwm, and ensured that the motors went straight with those increased pwm values. This allowed us to successfully completed the course in a time of 20.80s. At the end of the course, our robot did detect the stop, but staggered off the crossed black tape that served as a “stop!” indicator, and started forward slightly until it encountered the gray floor. Although this qualified as a course completion, we rewrote the code with an infinite ‘while’ loop at the stop to ensure that the arduino did not start again after detecting the stop indicator, but we did not have an opportunity to test this new code. We were slightly disappointed with our performance at race day, as the motors were slower than previously tested, so our time was not as good as we wanted. However, we still completed the course, which was initial objective. In that sense, we were successful.

Conclusions and Future Work

We were successful in building a mobile robot that could navigate the given course. Unfortunately, we were unable to successfully build a wheel sensor to detect when the robot stopped. The most valuable thing we learned building the mobile robot was how to turn a schematic into an actual circuit. In the beginning, the schematic was extremely daunting. However, by focusing on building the schematic one circuit at a time, we were eventually able to correctly convert the schematic into the circuits needed for the robot. After this experience, building circuits from schematics will be much easier. We learned how to build a transistor circuit and maximize its response; we also learned to always use a dropping resistor in an LED circuit.

If we had more time we would complete the wheel sensor, and attempt to integrate the wheel sensor into our pathfinding code to give our robot additional power whenever is stuck while following the path. We would also try to increase the speed of our robot, and test to find the maximum speed (pwm to motors) at which our code could navigate the course. We would first test these by running the given course and monitoring the robot’s progress and ensure that it completes the course (due to the fact that the robot seems to occasionally but consistently have its motors stuck, it would be capable to testing the wheel sensor code as well). We would then attempt to test on a variety of different courses and see if our code could run more difficult courses, such as those with tighter turns or more overlapping paths. If not, we would improve the code to make our robot more effective at navigating paths, while still maximizing the possible speed and retaining wheel sensor capabilities.

References

[1] PighiXXX. “Arduino Nano Pinout Diagram”. Feb 07, 2013. [Online]. Available: http://www.pighixxx.com/test/wp-content/uploads/2014/11/nano.png. Accessed on: Nov. 3, 2017.

[2] Fairchild Semiconductor, “TIP120/TIP121/TIP122 NPN Epitaxial Darlington Transistor,” TIP120 datasheet. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/TIP120.pdf. Accessed on: Nov. 10, 2017.

[3] *Arduino IDE*. Arduino, 2017.

[4] Arduino, “Language Reference”. 2017. [Online]. Available: https://www.arduino.cc/reference/en/#page-title. Accessed on: Nov. 10, 2017.

[5] B. Jeremy, *Tutorial 04 for Arduino: Analog Inputs*. Jan. 24, 2011. [Online]. Available: https://www.youtube.com/watch?v=js4TK0U848I&t=665s. Accessed on: Nov. 10, 2017.