Analog Plinko Robot

An Electronics Lab Robotics Design Report

Sarah Brown
College of Engineering
University of Oklahoma
Norman, Ok

I. Introduction

A. Overview of Competition

The 2020 electronics lab robotics challenge focused on designing a robot that would be able to travel down a hallway in Carson. The robot would travel down the corridor by "bouncing" as it completed turns between the two walls with the goal of making as many turns as possible. After reaching a marked distance down the hall, the robot would be switched from automatic mode to a manual controlled mode. In a manual control state, the robot would be steered over a bonus target and driven into one of several destination slots. At the end of the run, the robot would then shut off its motors and blink a LED signaling completion of the course. Due to the Coronavirus pandemic, the competition was changed to a design report to be built and simulated using Multisim.

II. Chassis

As the robotic challenge focuses on maximizing the amount of turns in a limited area, the ideal robot chassis for this project should focus on making the turn radius of the robot as small as possible. Due to this requirement, a small robot with two wheels with motors and a caster wheel in the back optimizes the turn radius while still providing a simple mechanical solution [1]. This chassis would provide a solid basis and would only require minor mechanical changes to add additional space for electrical components.

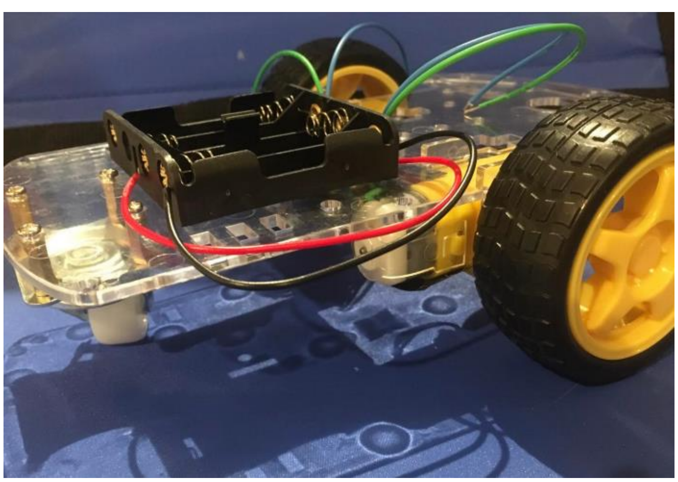

Figure 1: Chassis obtained for robotics design challenge. Image shows a side view of the robot with two front wheels and a small caster back wheel.

III. Check Off One

Check off one focuses on creating a way for the robot to be turned on without starting to move. Instead the robot should remain stationary until a momentary button is pressed and released. This challenge can be solved by creating a state machine. With the use of a state machine, the robot can progress through different operations based on certain triggers. There will be four main states for this state machine, these states can be seen in figure 2.

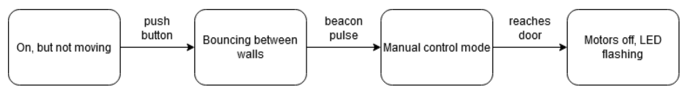

Figure 2: Different states of a state machine used to control robot behavior.

Based on experiments performed in the 555 lab during this semester, a basic state machine can be created by connecting 555 chips together in bistable mode and using them as basic SR latches. Shown in figure 3 is the main state machine. This state machine is started with a push button (labeled run in the schematic) which triggers the bounce state. With the use of NMOS logic and BS170s, it is possible to invert the output of the bounce and manual states and use the output of one state reset the other state (figure 4). Once the blinky state is entered, there is no need to exit it as it occurs at the end of the course. With the ability to toggle different states, the robot can proceed on its path down the hallway.

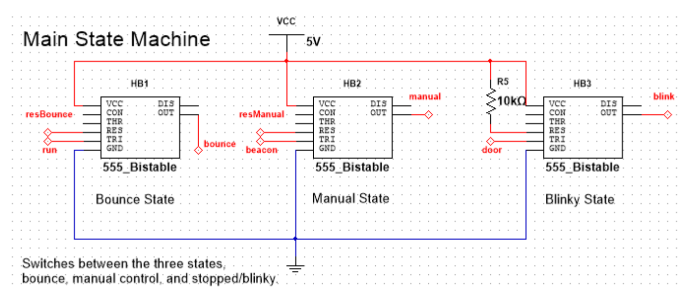

Figure 3: A subcircuit used to switch between different robot states.

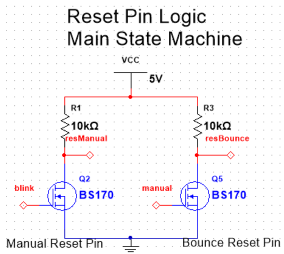

Figure 4: Reset logic for the bounce and manual states.

IV. Check Off Two

Check off two focuses on being able to detect a wall in front of the robot, turn towards the other wall, and repeat as it travels down the hallway. To detect walls an IR transmitter (a LED) and an IR receiver are used together. When the robot is a certain distance away from the wall, the IR light bounces off of the wall and seen by the IR receiver. The exact distance to trigger the IR receiver would have to be tested in a real-world situation. The set up is very similar to the IR LED and receiver set-up is similar to the one used in the soldering lab this semester. For the purposes of simulation, the push button named wall (triggered by key F) is used to simulate detecting the wall. The pulse generated by the IR receiver is used to trigger different states in another state machine. The different states used to perform the bounce action are shown in figure 5. The 555 chips are configured in the bistable mode to be used as SR latches. Using SR latches allows a circuit to store a certain value or configuration until reset. The reset pins for the four 555 chips are tied to the next state in the state machine which then forces the previous state to reset when the next one is triggered. The trigger pins are controlled using logic based on the previous state and input from either the IR receiver or a pulse generated when the left/right turn state is completed. The drive left state is triggered when the bounce state machine first starts or when the turn left state ends (shown in figure 6). When the state machine is in the drive left state, the robot drives straight forward towards the left wall. When the robot gets close enough to the wall it triggers the turn right state. In the turn right state, the left motor moves forward and the right motor drives backwards resulting in the robot turning to the right. The turn length is determined by a 555-timer circuit configured in monostable normally off mode. The turn length would be determined experimentally by changing the period of the timer for the optimal turn time. Due to the original placement of the robot facing the left wall and the precision of the state machine the robot should be able to achieve a high number of "bounces" before reaching the manual portion of the course. This timer is started when the right turn state is triggered and when it ends it generates a pulse to trigger the drive right state. The drive right state and the turn left state function very similarly to the previous two states. All of these modes output signals that are processed through NMOS logic to control which direction the motors and the robot are going.

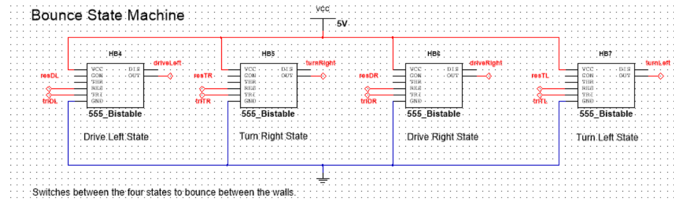

Figure 5: The bounce state machine that switches between driving straight forward and turning either right or left.

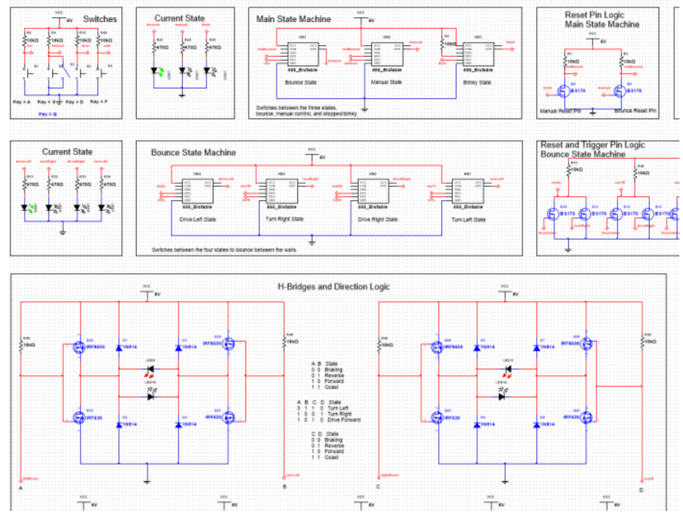

Figure 6: The robot driving forward while in the bounce state.

V. Check Off Three

Check off three focuses on manually controlling the robot and being able to drive it over a randomly placed target and park it near the door at the end of the hallway. On the rear of the robot chassis is an IR receiver and by using a beacon that emits a pulse of IR light it is possible to trigger manual mode of the robot. During manual mode the IR receiver on the front of the robot used for wall detection would be OR-ed together with the IR receiver on the rear of the robot to provide full converge while the robot is turning. Manual mode of the robot is controlled by another state machine as seen in figures 7 and 8. On the rising edge of a beacon pulse either the drive forward or the turn state (shown in figure 9) is activated. The two pause states serve only as waypoints to make sure that the state machine does not oscillate between driving forward and turning. By pressing and holding the beacon button (simulated with a press button for short presses and a switch for extended presses) it is possible to steer forwards and then transition into a turning state. By combining these two modes the robot can be controlled manually. With the addition of a push button on the front of the robot when it runs into the door at the end of the hallway it will change to the blinky state. In the blinky state the robot will turn off the motors and blink a LED signaling the end of the run.

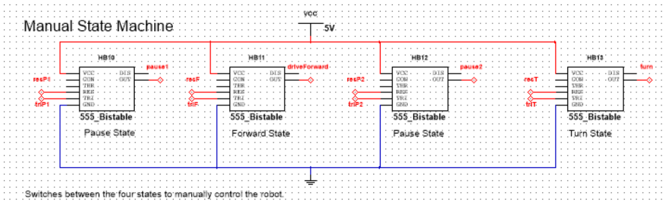

Figure 7: A manual state machine designed to alternate between pauses states where nothing is output, a forward state, and a turn state.

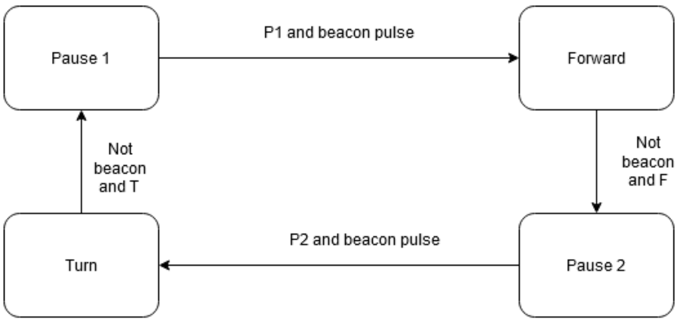

Figure 8: A figure showing the mapping between the different states in the manual state machine.

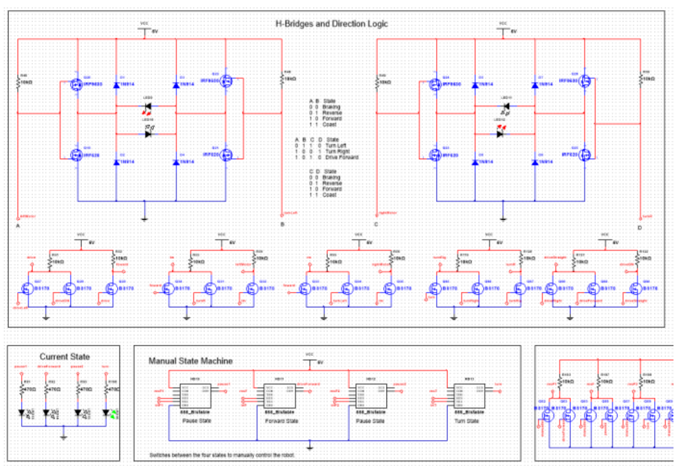

Figure 9: The robot shown in the turn state while under manual control.

VI. Overall Design

A. H-Bridges

The robot design used to complete this project features two motors and the motors are controlled using H-bridges. These H-bridges are designed using the same style as the one built in class in lab. The H-bridges output is controlled using NMOS logic to determine the intended robot direction based on the given state from either the bounce state machine or the manual state machine. To minimize the turn radius of the robot, turning is implemented by driving one motor forward and the other motor backward. The H-Bridges control the separate motors in three modes: coasting, forward/reverse, and braking. The H-bridge design and NMOS is shown in figure 10.

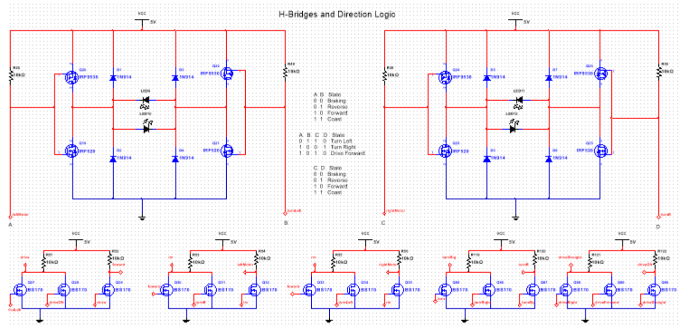

Figure 10: Two H-bridges and necessary logic to control two motors based on inputs from various robot states.

B. Beacon

The beacon used to control the robot in manual mode is the same beacon that was created during the soldering lab. The schematic for this beacon can be seen in figure 11. The beacon operates on a 38kHz frequency which is useful for avoiding interference from ambient lighting.

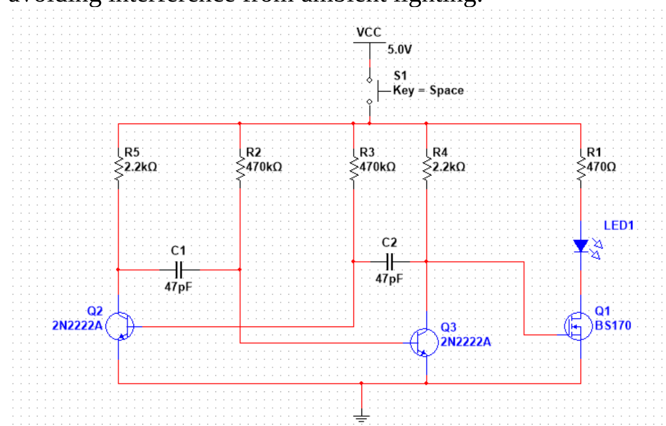

Figure 11: Schematic of beacon used to control robot in manual mode.

C. Pulse Generators

Throughout the design of this robot pulses are needed to trigger 555-timer circuits, but the input signal used is in a level form. To convert from a level to a pulse a resistor-capacitor circuit is used. The circuit shown in figure 12 was based on a level to pulse convertor from *Engineers Garage* [2].

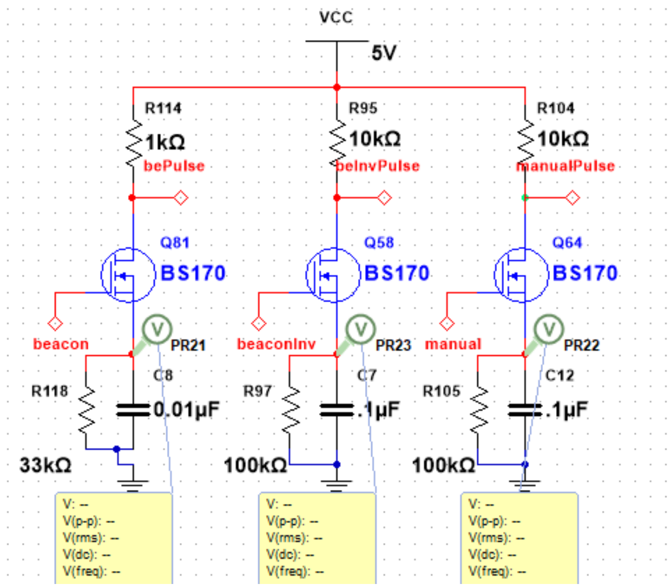

Figure 112: A level to pulse generator used throughout the robot project to trigger 555-timer circuits.

VII. Issues and Challenges

The main issues and challenges faced were regarding Multisim. Some items that were necessary to complete this challenge proved difficult to simulate. This mainly was due to Multisim viewing holding down a push button switch as multiple button presses, to combat this issue I added a toggle switch for the beacon. This switch is used to drive the robot in manual mode. Whenever pulse generators were implemented in this design they did not respond in real time and took a long time to stabilize which caused some simulation issues. However, by changing the values of the components composing the pulse generator it was possible to create a circuit that would simulate. To ensure that it simulates correctly long pauses must be taken to allow the capacitors shown in figure 12 to stabilize.

VIII. Conclusion

Given the circumstance surrounding the robotics challenge, researching, designing, and simulating it is a realistic and flexible solution to a unique problem. However, some issues were posed due to Multisim being unwieldy for large projects. These issues were mainly focused on troubleshooting and simulation time. However, even though there were different issues tackling a project in this format, I still learned a lot and enjoyed learning more about how to design an analog robot.

References

[1] https://www.amazon.com/gp/product/B01LWYUQPH/ref=ppx_yo_dt_b_asin_title_o09_s00?ie=UTF8&psc=1 Link for robot chassis

[2] A. Sammuddin, “Level to Pulse Converter,” *Engineers Garage*, 12-Jul-2019.[Online].Available: https://www.engineersgarage.com/contributions/level-to-pulse-converter/. [Accessed: 30-Apr-2020].