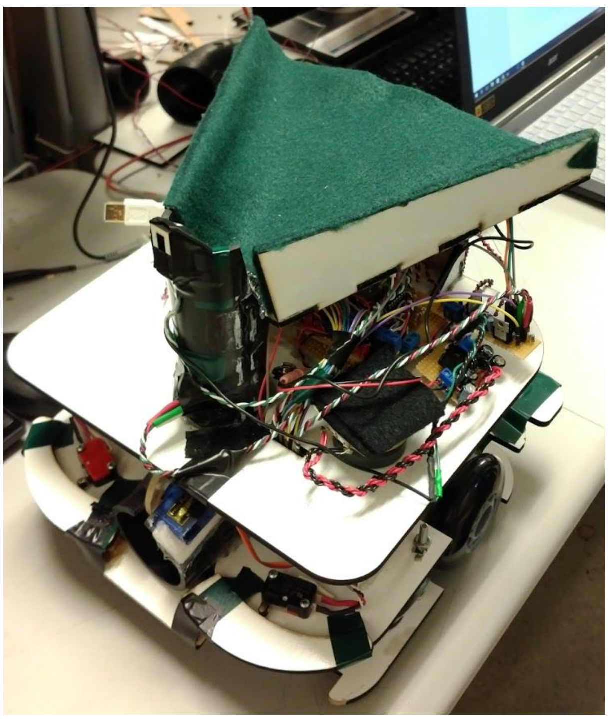
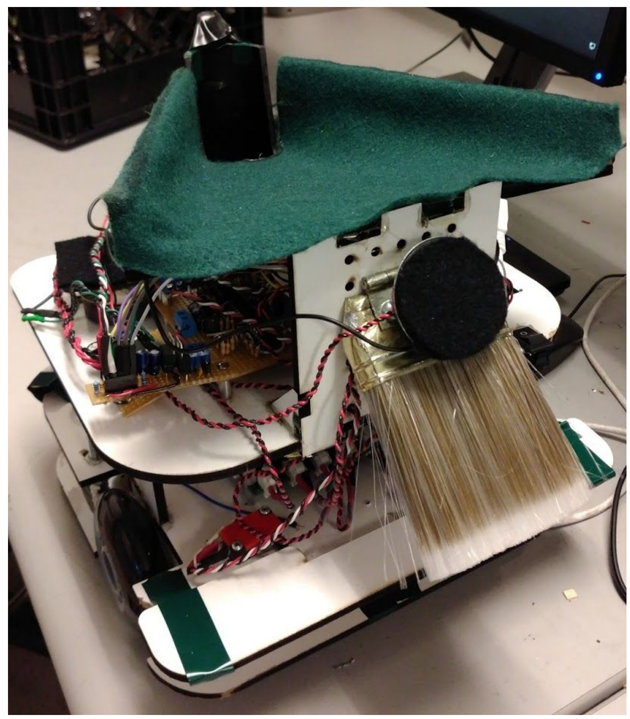
CMPE118 Mechatronics Final Project Report

Instructor: Gabriel H. Elkhaim

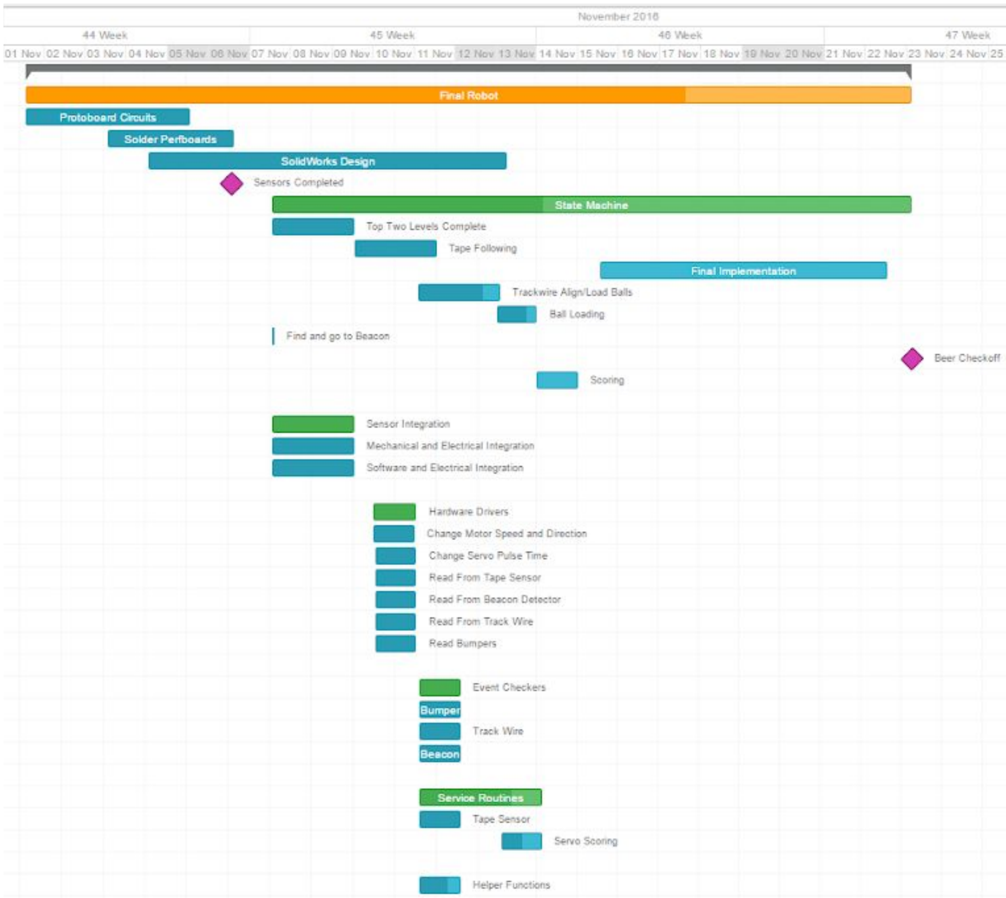
Team 16 - Slugo de Golfo:
Joshua Pena
Joshua Passmore
Julia Warner

Featuring the Robot:
Will Smith

Introduction:

Initial Planning:



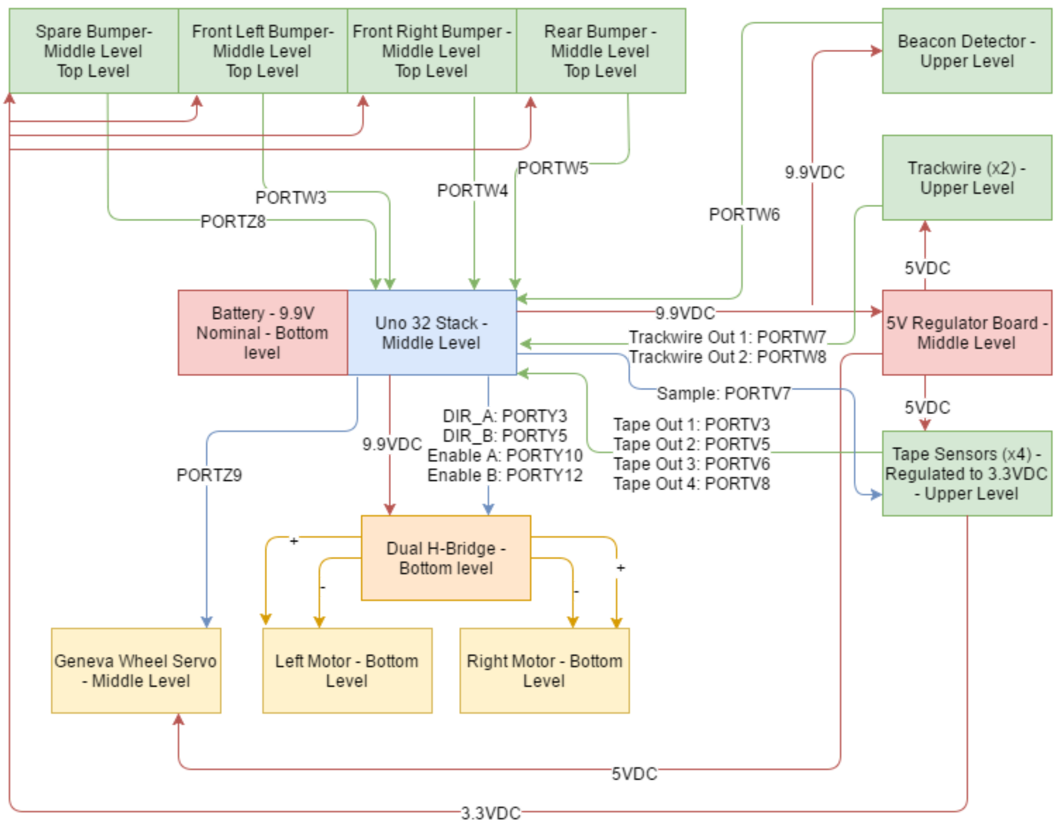


Figure 1a and 1b: Planned interconnects and sensors for the robot and Gantt Chart Planning schedule. Some additional connections were necessary but the final product is very similar to this block diagram except two additional tape sensors were added, as well as a speaker system (that ran on a separate 9V battery).

Electrical Development:

a) Trackwire Sensor Development

The trackwire sensor was the same device developed in lab 1 but now needed to be soldered onto a perfboard. In overview the trackwire resonates at a harmonic frequency of 24-26 kHz based on an inductor-capacitor tank circuit and through amplification, peak detection, and comparison a clean on/off signal is obtained that indicates whether a trackwire signal has been detected. For the purpose of this lab the range was set at approximately 4-5 inches (without obstruction) since the signal needed to be sensed through MDF and a sheet of plywood. This range allowed the inductors to be placed very close to the ground and still be precise for the desired detection application. In the interest of ensuring the design was correct the components were laid out on a protoboard and functionality was verified. This period of verification was where all the problems were caught and was the most frustrating portion of the trackwire development. Moving to the perfboard was easy and two trackwires were created on the same board. The compact form factor of the trackwires made incorporation into the robot easy and quick. Power was supplied to the trackwire by a power distribution board with a 5V Linear Dropout Regulator (LDO) that was reverse polarity protected. Final design can be seen in the images below:

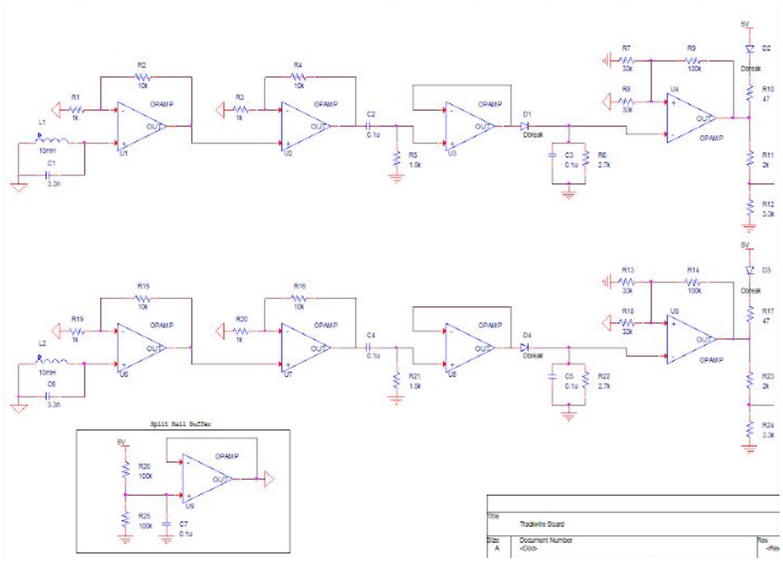
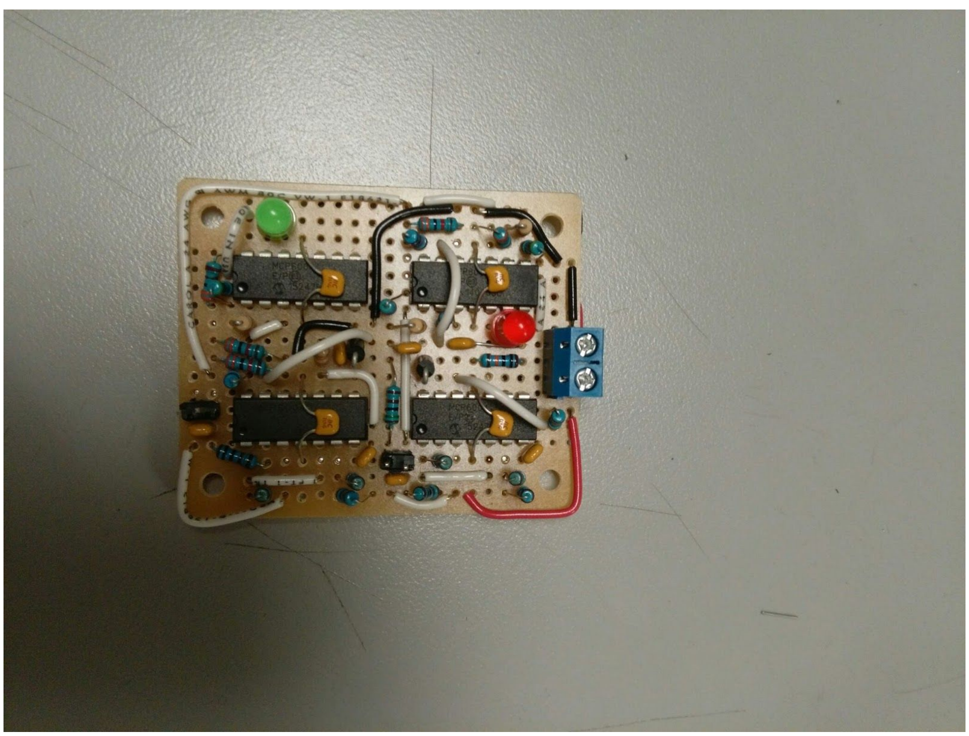


Figure 2: Trackwire circuit diagram for the two trackwire board. Note that there are two buffer stages (one for each trackwire circuit) that should be inserted between the final op amp stage and the LED voltage divider. Also note the use of an op amp, rather than a comparator chip for the comparator portion of the design.



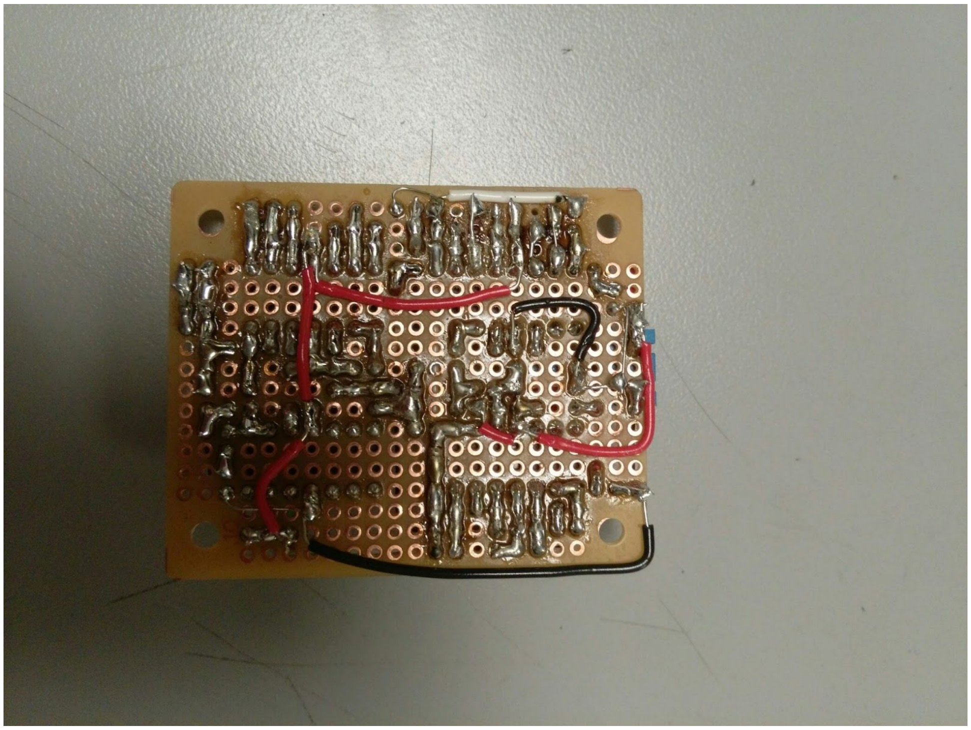


Figure 3 and 4: Actual trackwire circuit implemented on the perfboard with top and bottom views. The two LEDs correspond for the two separate trackwires. White wires show the signal path between various stages and where there was not enough space to directly solder between connected points.

b) Tape Sensor Development

The tape sensor was a new sensor for the final project and required development. In order to understand what needed to be done with the TCRT5000 IR LED/Phototransistor pair, the datasheet was consulted and pre-built solutions were examined to see what values of pullup resistors were used to ensure the correct operation of the sensor. A NPN Power Darlington was also needed to be able to control all the TRCT5000s simultaneously from a single impulse sent by an Uno pin. The design that ultimately worked best was using a 100kΩ resistor between the phototransistor collector and 3.3V, a 330Ω resistor between the anode of the IR IED and 3.3V, tying the emitter of the phototransistor to ground, and controlling the input to the cathode of the IR LED using the collector of an NPN power darlington, where the emitter was tied to ground and the base was controlled through a signal from an Uno pin which was current limited using a 3.3kΩ resistor inline with the uno signal. 3.3V was supplied by a 3.3V LDO that was onboard with reverse polarity protection. Development of the tape sensor was easy for the initial board but additional tape sensors were determined to be necessary for added functionality. The most difficult part of the development was creating the 4-wire bundles for the TCRT5000s and creating the ribbon cables for input/output to the perfboard. The final design can be seen in the two images below:

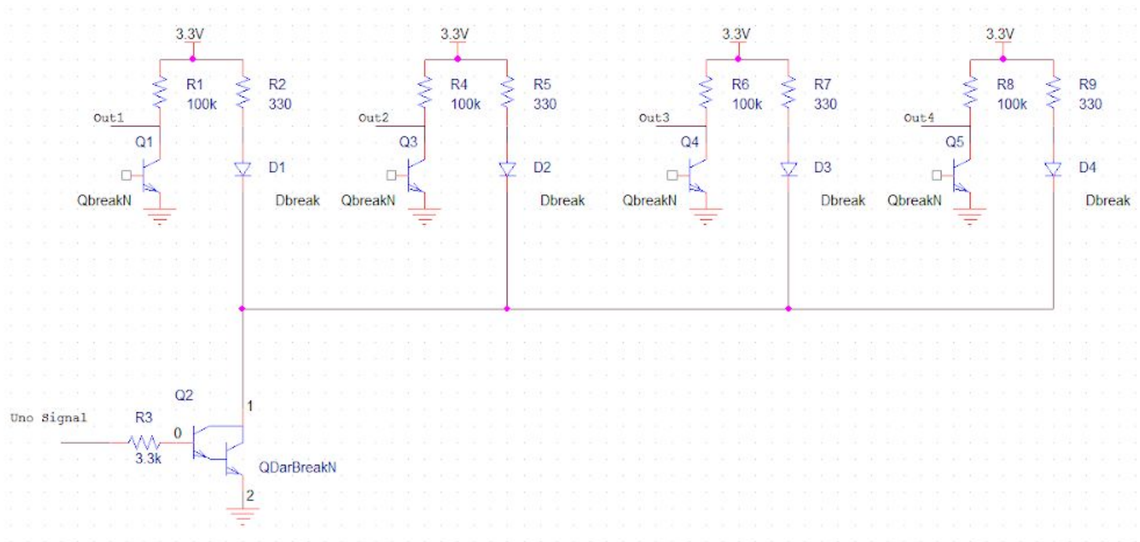


Figure 5: Tape sensor circuit diagram showing the single input controlled switch of the NPN power darlington and the four output signals that are sent to the Uno stack

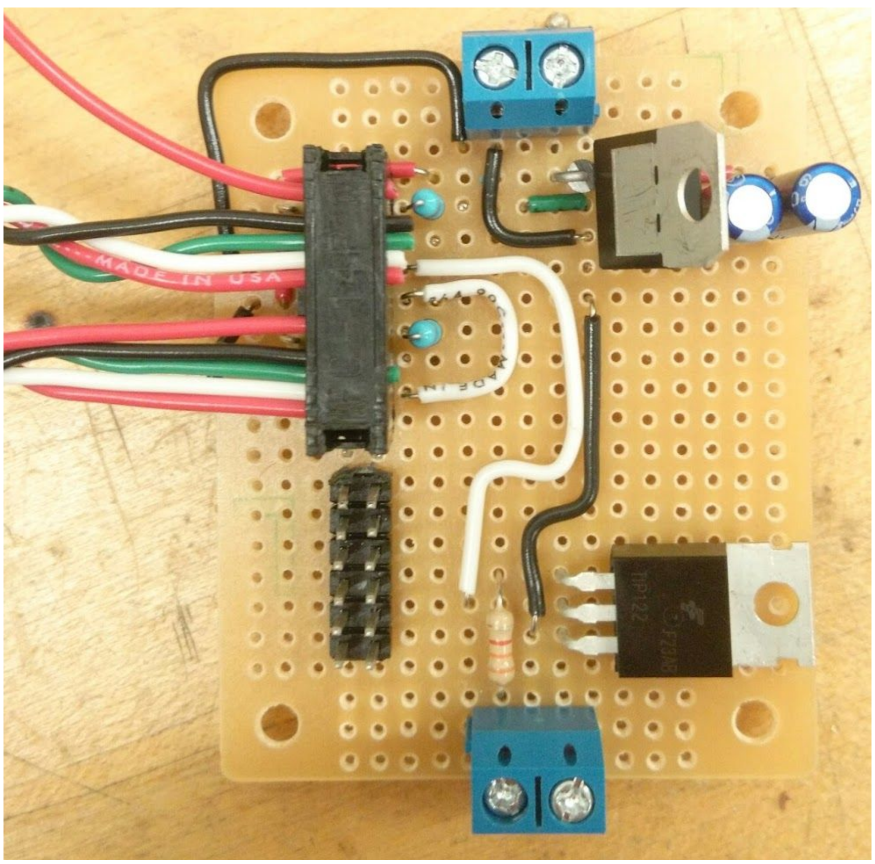


Figure 6: Actual perfboard implementation of the tape sensor board. Additional boards were later added to give more tape sensors but this is the general layout using a TIP122 power darlington and a 3.3V LDO with reverse polarity protection

c) Bumper Sensor Development

The bumper sensors were developed in conjunction with another team. The general design was determined to work best when the configuration was set with 3.3V attached to the NO terminal of the SPDT switch, ground tied to the NC terminal of the SPDT switch and the output tied to the common of the SPDT switch, with a current limiting resistor and LED indicator to ground. The design presented no problems and the design was highly modular, with the limit switches able to be replaced through screw terminals. Everything worked on the first try so there were no difficulties to report. Finding a place for the perfboard on the robot was a bit tricky but the ultimately the board was stacked on the edge of another board. The design for the bumper sensors can be seen in the two images below:

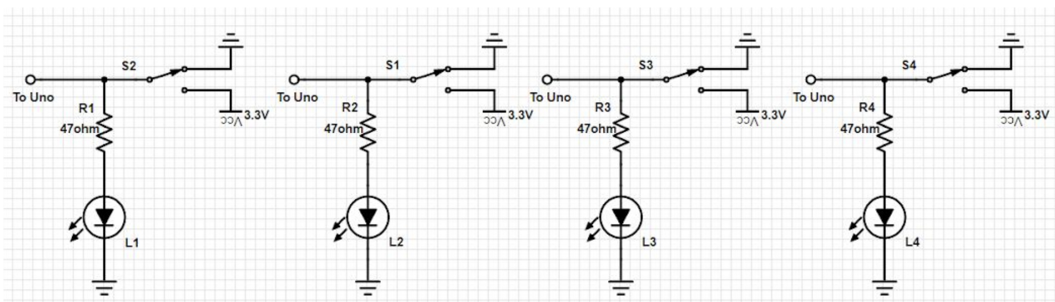


Figure 7: Circuit design for the bumper sensor board. The configuration of the board is using SPDT switches that are NC to ground and when depressed transition to NC to V_{CC} = 3.3V. Depressing the switch also illuminates an LED corresponding to the desired switch.

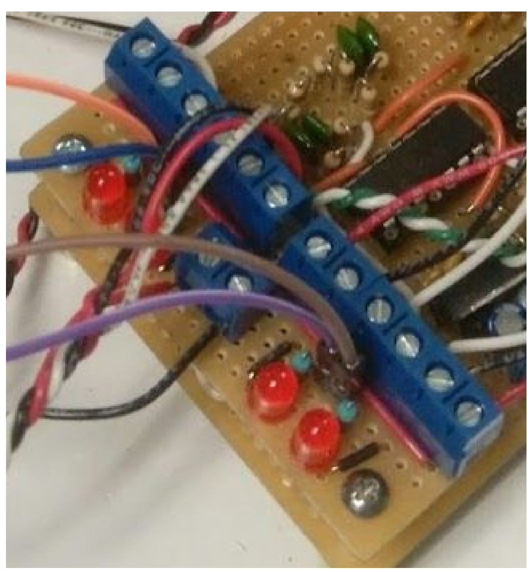


Figure 8: Actual perfboard implementation of the bumper sensor board when it was mounted on top of the beacon detector. Note the screw terminals for all the bumper terminals and the four LEDs corresponding to the different bumpers.

d) Beacon Sensor Development

The beacon sensor was already developed from lab 2 and was utilized in its existing form. It is a 6th order Chebyshev bandpass filter with cutoff frequencies of 1.9 kHz and 2.1 kHz. It has 3 gain stages and an on board 5V voltage regulator. A connector is used to make it easier to put in different sized wires and an LED is used to show when the output signal is high or low. The actual perfboard design was done in Eagle. These are the basic component code values.
C1 = 104; C2 = 104; C3 = 104; C4 = 100; C5 = 100; C6 = 100; C7 = 100; C8 = 100
; C9 = 100; C10 = 104; C11 = 104; C12 = 104; C13 = 104; C14 = 104
R1 = 103; R2 = 103; R3 = 474; R4 = 102; R5 = 102 -> 105 to C1; R6 = 103;
R7 = 102 -> 105 to C2; R8 = 103; R9 = 102; R10 = 103; R11 = 474; R12 = 100;
R13 = 100; R14 = 105; R15 = 104; R16 = 474; R17 = 562; R18 = 100; R19 = 105
; R20 = 105; R21 = 105; R22 = 474; R23 = 474; R24 = 474; R25 = 474; R26 = 105
; R27 = 272; R28 = 202; R29 = 332; R30 = 470; R31 = 103; R32 = 103; R33 = 474
; R34 = 474; R35 = 470; R36 = 470; R37 = 474; R38 = 474; R39 = 122

To be more compact, 3 MCP6004's were used instead of 3 MCP6004's and a dedicated comparator. The extra Op-Amps mean that a fourth chip is not needed. Sockets were also used to make sure that the chips are replaceable if something goes wrong.

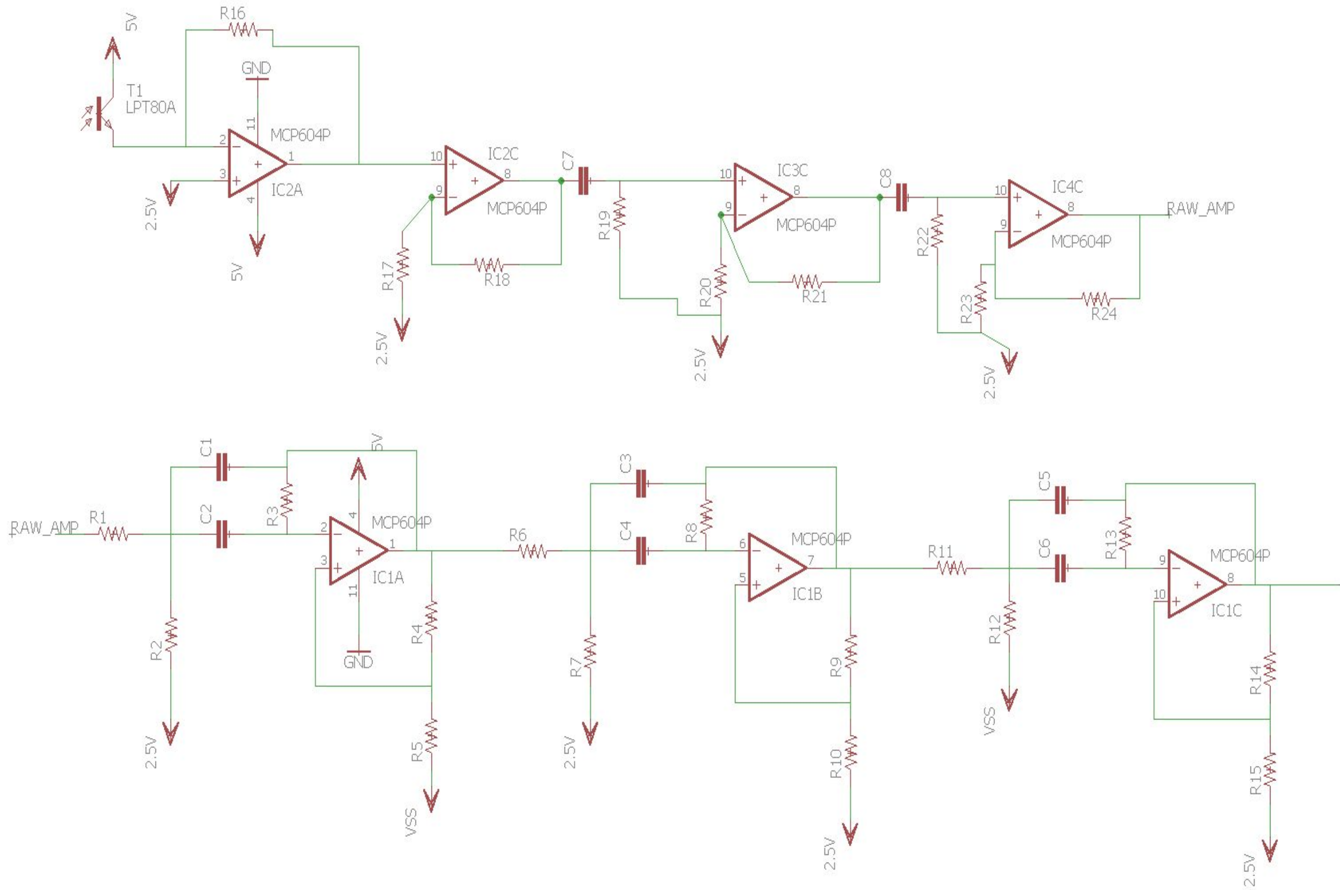


Figure 9: Beacon Filter Schematic

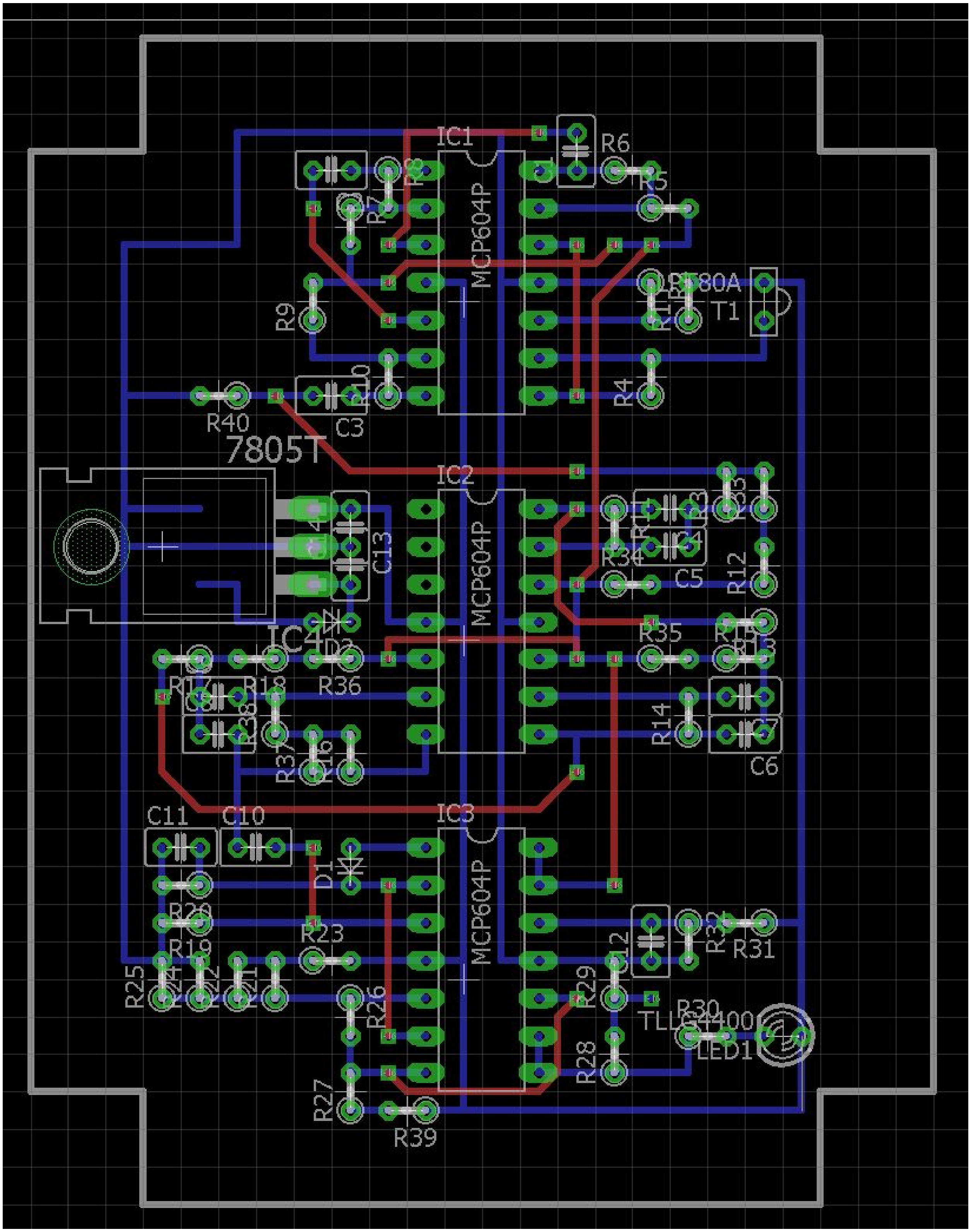


Figure 10: Perfboard Layout for Beacon detector (R40 is no longer used)

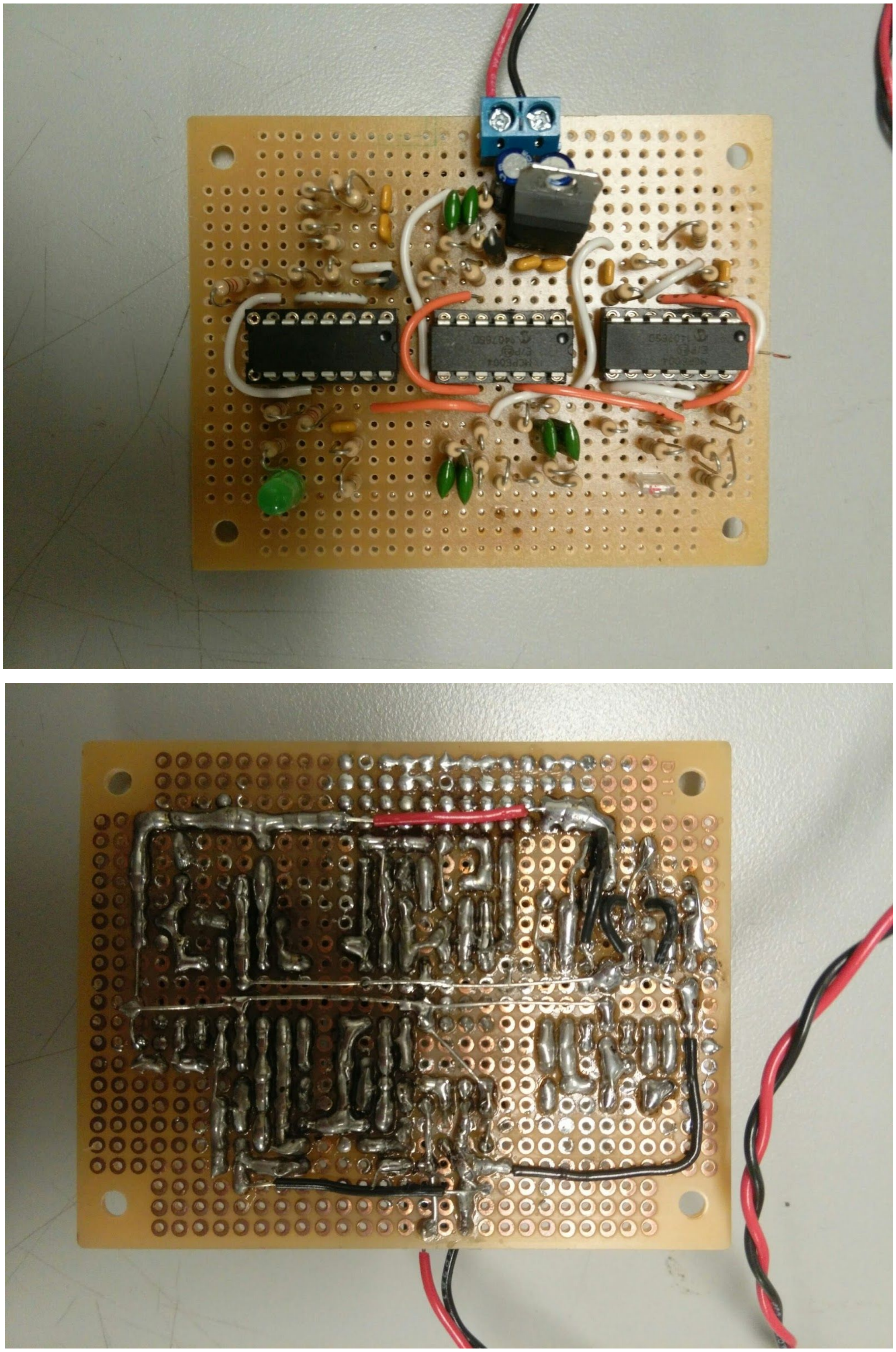


Figure 11 and 12: Actual implementation of the beacon sensor board with top and bottom views.

e) Speaker System Development

The speaker system was a secondary project for fun that was added at the tail end of the project. In order to create the speaker amplification circuit an audio amplifier chip (the LM380) was used as a 14 pin IC package. The capacitors needed as charge repositories were 470μF, which were ultimately composed of two 220μF capacitors in parallel (which gave 440μF capacitance). This was a reasonable deviation from the expected capacitance. Audio output was through two speakers, one pointed upward and the other pointed backward. The speaker facing upward was a 3W 8Ω speaker that covered more bass tones and the backwards firing speaker was an ½ W 8Ω speaker that had less bass capabilities. Volume of the speaker circuit was accomplished through use of a potentiometer that ranged from 0-10kΩ. Audio input to the circuit was through auxiliary cord to a 3.5mm headphone jack. The auxiliary connector was soldered by separating the voltage and ground lines from one another (since the auxiliary cord comes as a coaxial line with small gauge wire for ground surrounding an inner, insulated hot wire strand. The first attempt at the auxiliary configuration was unsuccessful and resulted in destroying the headphone jack and microphone in one of the team members phone. Fortunately the team member had a backup phone and was able to survive until he ordered a new phone.

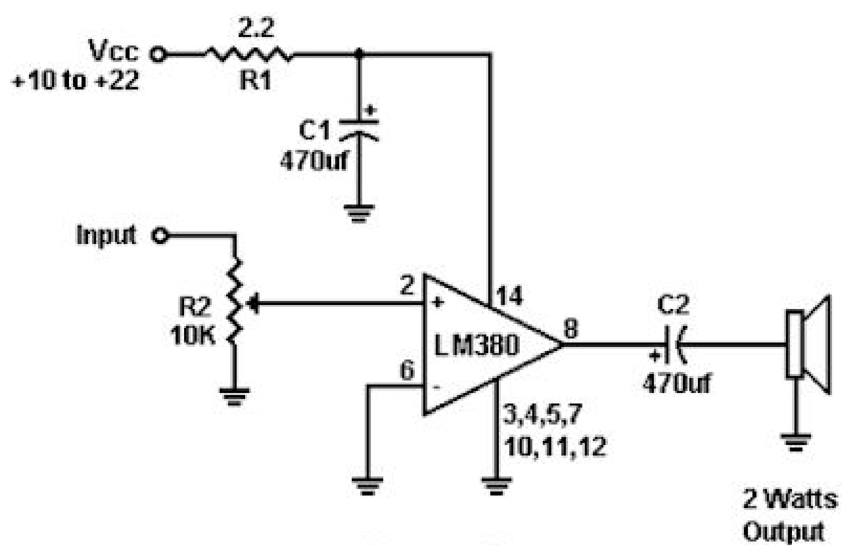
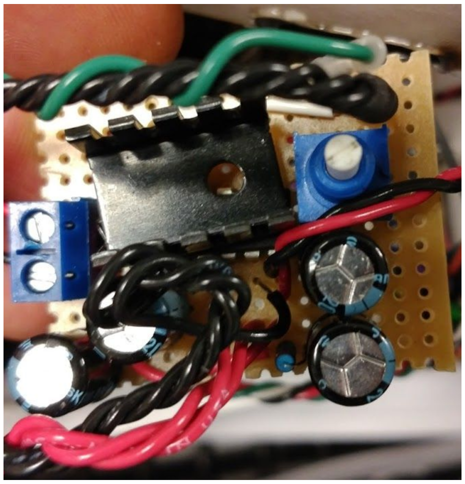


Figure 13: Credit for design goes to http://www.rason.org/Projects/icamps/icamps.htm for the use of the LM380 chip as a single stage audio amplifier. Note the speakers were not capped at their maximum power since the output could only source 2 watts.



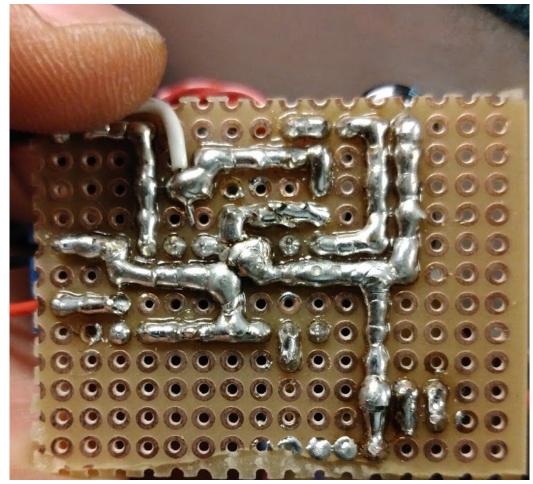
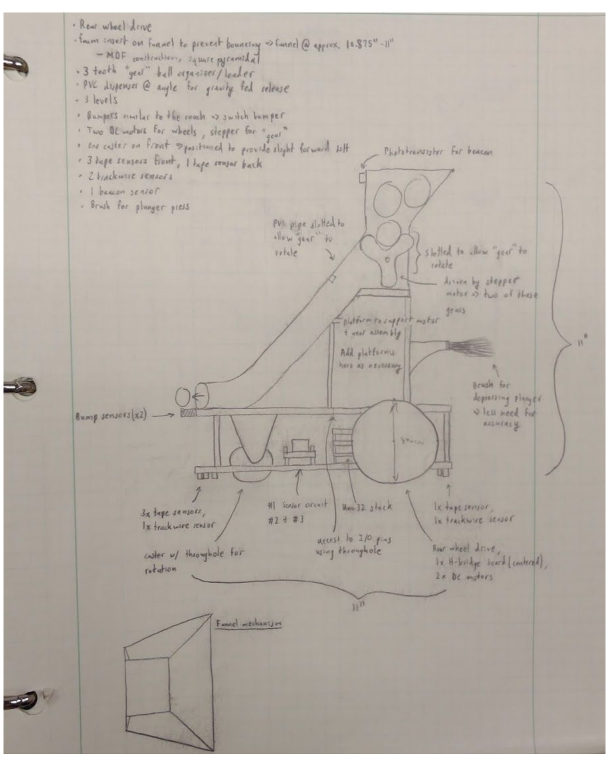


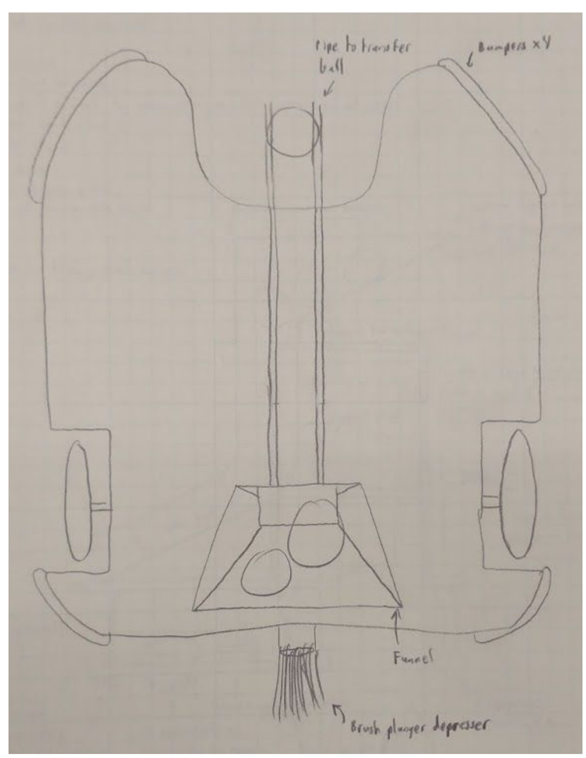
Figure 14 and 15: Actual perfboard implementation of the speaker system circuit. Note the LM380 needed a heat sink when music was played for extensive durations since the chip would begin overheating , as classified by touch test.

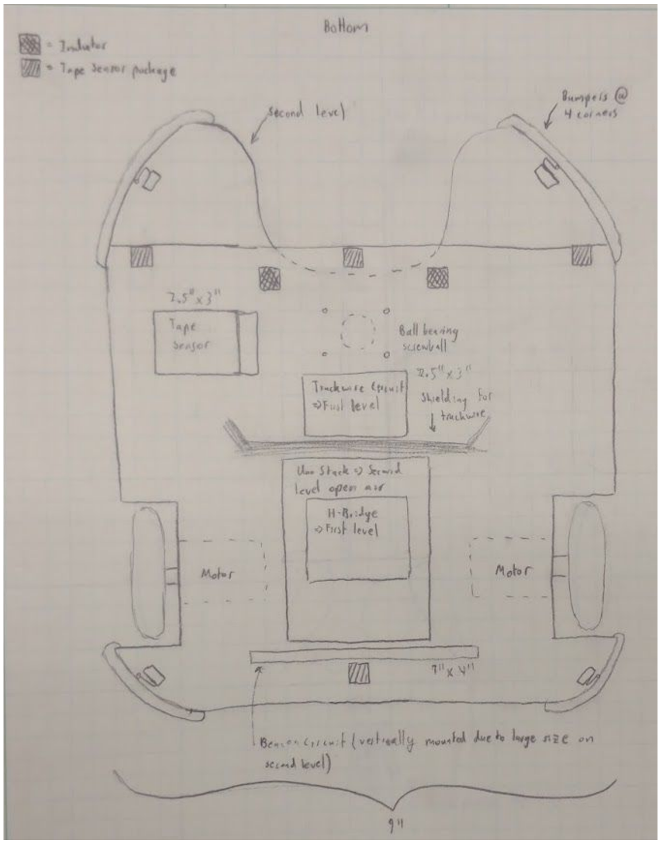
Mechanical Development:

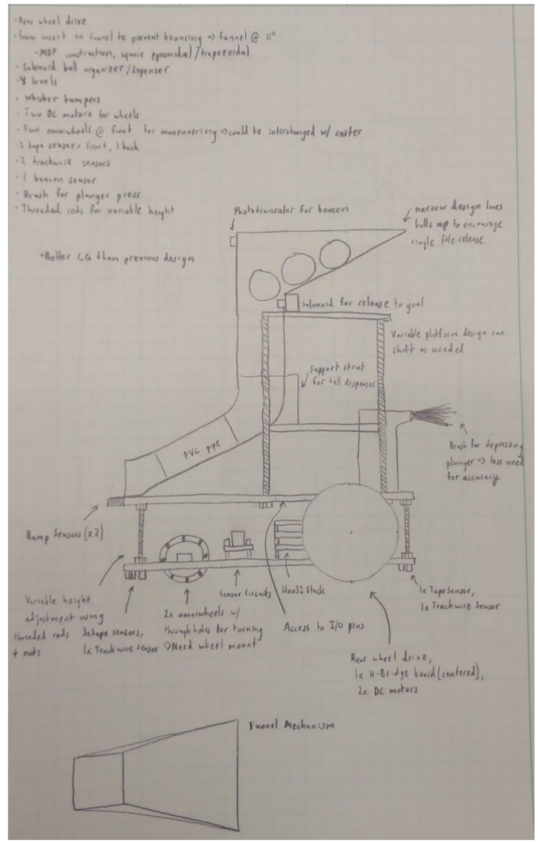
a) Initial Designs

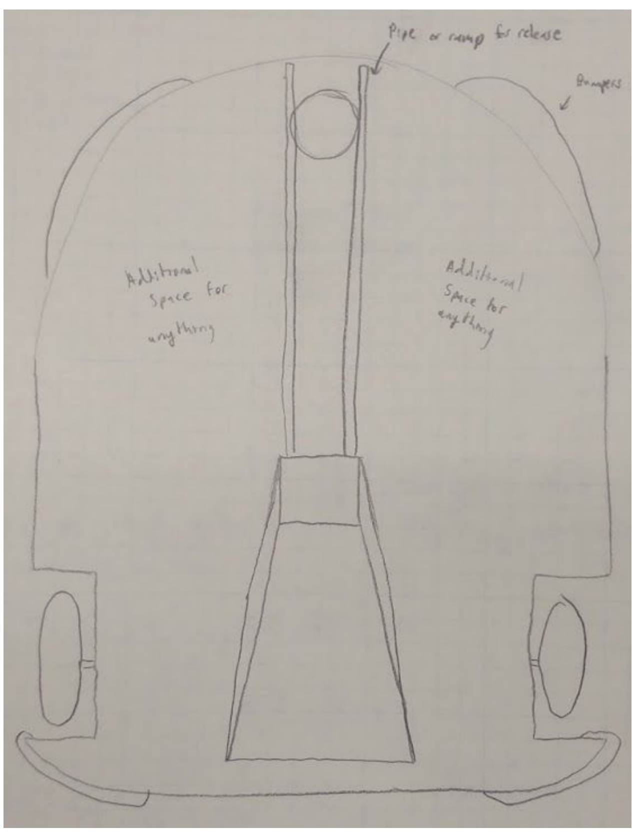
For the initial designs three different schematics were drawn with different configurations/implementations with respect to both mechanical design and software implementation. The three design covered the different wheel placement (rear wheel, center wheel, and front wheel, as well as different ball release mechanisms, different structural compositions, and sensor placement. The designs can be seen below in their initial form.

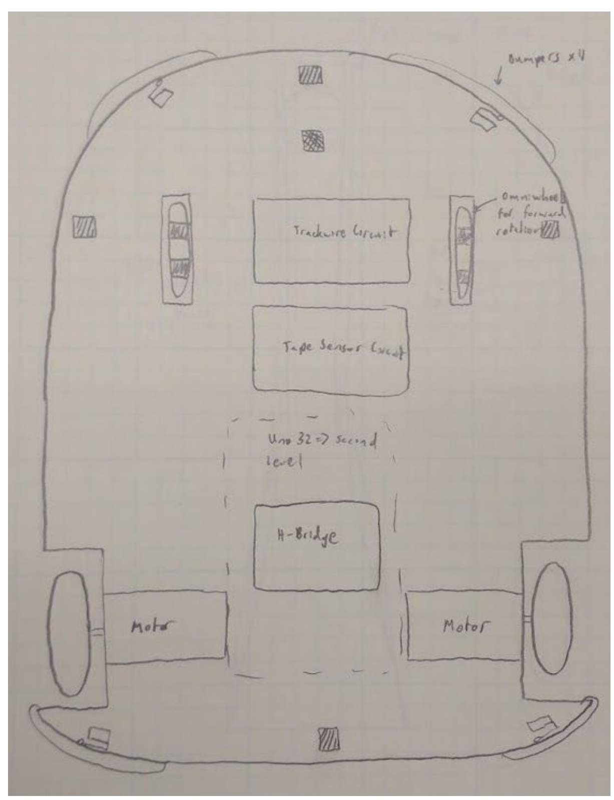




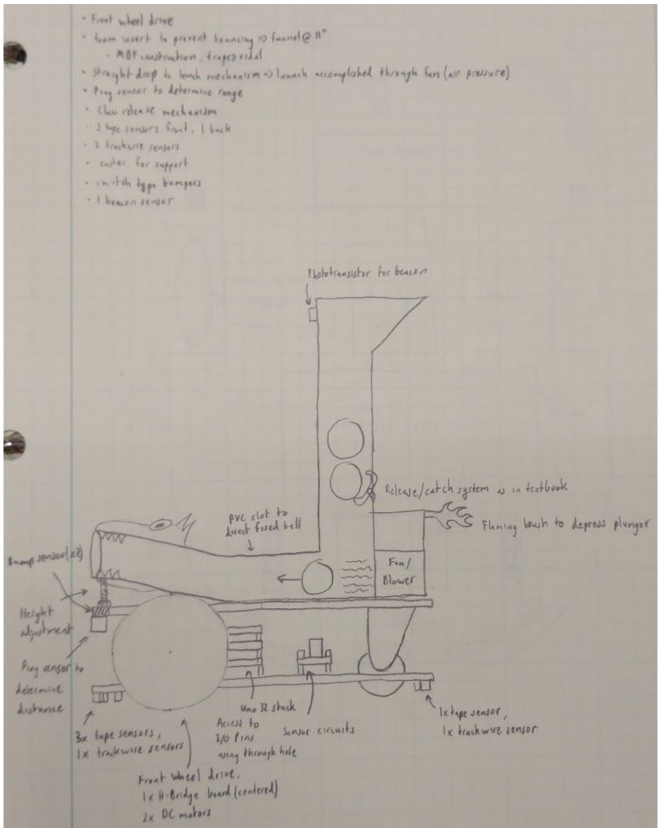

Bottom
= Indicator
= Tape Sensor package
Second level
Bumpers @ 4 corners
2.5" x 3"
Tape Sensor
Ball bearing screwball
Trackwire Circuit => First level
2.5" x 3"
Shielding for trackwire
Uno Stack => Second level open air
H-Bridge => First level
Motor
Motor
7" x 4"
Beacon Circuit (vertically mounted due to large size on second level)
9"

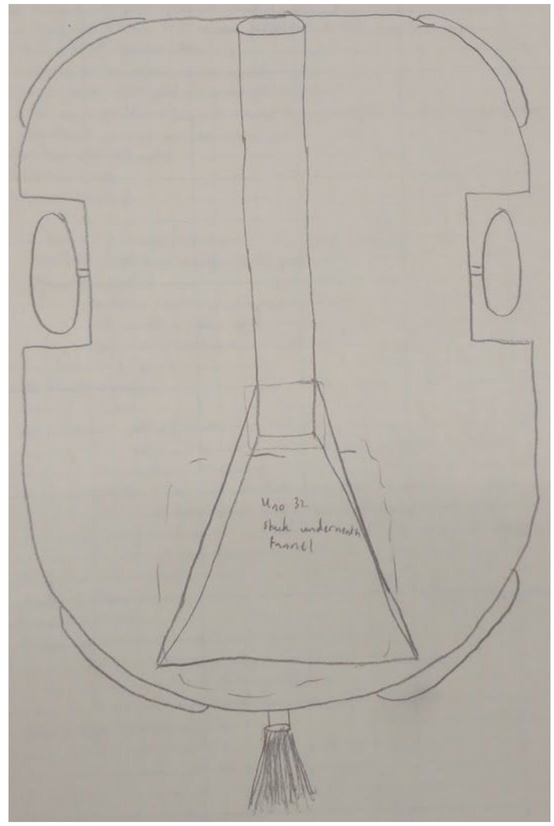

· Rear wheel drive
· Foam insert in funnel to prevent bouncing => funnel @ 11"
- MDF construction, square pyramidal/trapezoidal
· Solenoid ball organizer/dispenser
· 4 levels
· Whisker bumpers
· Two DC motors for wheels
· Two omniwheels @ front for maneuvering => could be interchanged w/ caster
· 3 tape sensors front, 1 back
· 2 trackwire sensors
· 1 beacon sensor
· Brush for plunger press
· Threaded rods for variable height
Phototransistor for beacon
narrow design lines balls up to encourage single file release
*Better CG than previous design
Solenoid for release to goal
PVC pipe
Rear wheel drive, 1x H-Bridge board (centered), 2x DC motors
Access to I/O pins
Funnel Mechanism


Pipe or ramp for release
Bumpers
Additional Space for anything
Additional Space for anything


Bumpers x4
Trackwire Circuit
Tape Sensor Circuit
Uno 32 => Second level
H-Bridge
Motor
Motor

b)





Figures 16 - 23: Initial designs for possible robot configurations. These designs covered a range of different drivetrains, launch mechanisms, construction tactics, and sizes.

Solidworks Components/Model

Once the initial designs had been drafted work began on the Solidworks design. The initial Solidworks components did not survive the iterative process since design considerations changed once the goals were released with their ribbed design. This led to the robot moving from a two level design to a 3 level design to accommodate the Uno stack (the control center), dual H-bridge (for bidirectional control of the motors), all the sensor boards, the ball dispersing method, and ball collection method. The redesign was a major setback since it necessitated redesigning the top two levels of the robot after the design had been finished. However, it was worth the effort since the robot performed exceptionally well on ball release compared to other robots. Final implementation was as follows.

The bottom level of the robot was the lowest part of the robot, with the motor mounts sitting on top of this level. The wheels were attached to the motors at this level and two 75lb capacity ball casters were recessed through the bottom of the lowest level, forward of the rear wheel drive by several inches. Five tape sensors and the two trackwire circuits were also placed on this level while their respective perfboards were on the third level of the robot. The H-bridge was placed between the motor mounts. The front of the bottom level was constructed so that it could get closer to the goal by being smaller than the levels above it. There was also an added feature with the small rounded horn on either side that could be used to massage the goal into position if needed, as well as push goals that were on the outside edge out of the way so the robot could get past them.

The middle level was attached to the bottom level by a set of hinges in the rear of the robot that allow the robot to be folded open to work on the lowest level. Support was provided by the motor mounts and the level was prevented from opening by placing nuts on the long screws that secured the ball casters to the bottom level. The Uno stack and the battery were placed on the middle level, with through holes incorporated for wire pass-through to the bottom level. For the 90° elbow pipe used for ball dispersal a filleted slot needed to be cut to accommodate pipe placement.

The top level was secured to the middle level using two tab-in-slot pillars and two hinges on one side, again allowing access to the middle level. All the sensor perfboards, as well as the audio circuit, were placed on this level. A support structure was also added on this level to support the ball collection funnel, which was supported on the other side by the 1-½" PVC pipe that collected the balls. A hole precisely the size of the PVC pipe was placed in this level for the pipe to pass through to the middle level, where the 90° elbow was placed. A servo mount was supposed to be attached to this level but the final placement of components made this configuration impossible so the servo mount was eliminated in favor of placing the servo directly on the pipe elbow and machining the elbow to have a slot for the geneva wheel (crescent moon shaped ball release mechanism). As with the middle level there were was a pass-through for the wires to move from the top level to the middle and bottom levels.

Bumpers were designed for the robot but the best securement method the team could develop was using tape to secure the bumpers. The bumper shape was designed to be curved and pushable from a range of positions and the tape allowed the motion to remain free and usable from all angles the bumper was hit at. The robot incorporated three bumpers and determining the exact size, placement, and securement method was an extensive process. The most frustrating part of working with the Solidworks model was attempting to fork the design to a new iteration with a new name caused the design to become largely unusable since the assembly had dependencies to other components and as such would not allow something with the same components to be edited. Consequently, there is only one version of the robot, as opposed to the multiple versions that would have been ideal. The final design of the robot can be seen in the below figure, which is the assembled Solidworks model.

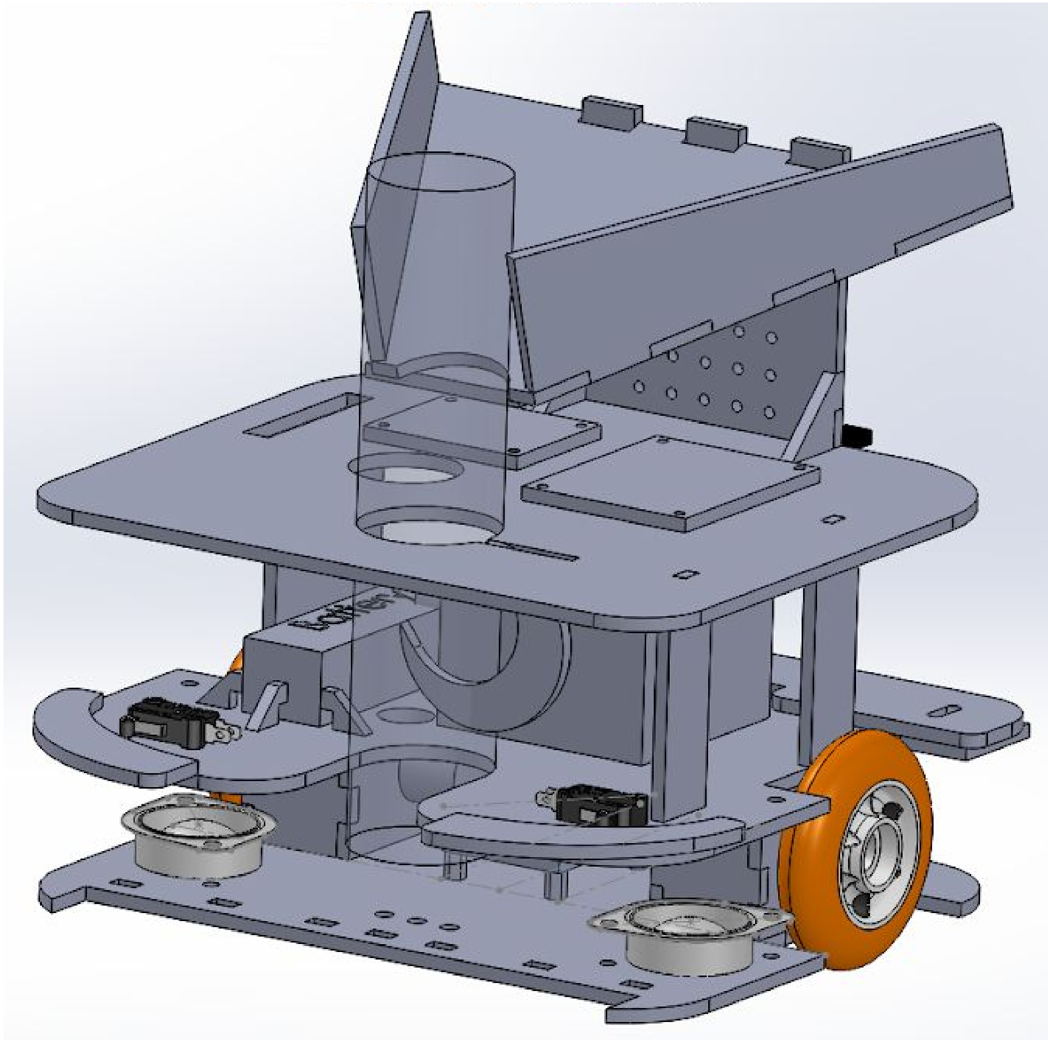


Figure 24: Final robot design as modeled in Solidworks. Note the large funnel, blocks for various perfboards, casters, wheels, battery, limit switches, and PVC pipe have all been modeled to exact scale to aide in the design of the robot.

c) Laser Cutting Components

All the components for the robot were created out of MDF and were cut with the laser cutter. Minor adjustments for things that had not been accounted for in the model, such as additional holes, slight sanding, and a slot for the remote power switch. Dealing with the laser cutter not fully cutting pieces and having to be set to speeds from 3.5 - 5 for MDF was frustrating, and additional cuts were needed to remedy pieces that did not properly cut when there was a time constraint for the laser cutter. The precision of the laser cutter was nice but the process of getting clean cuts was a somewhat painful experience.

d) Pushing Plunger

One part that continually changed throughout the course of development was the loading mechanism itself. Though we had intended to use a brush design from the start, how it was going to be fitted and whether it was the best choice were under debate whenever we got to it. We ended up getting to it much later than anticipated and by the time we got a brush, we soon realized some issued with the particular brush we got. Using a cheap plastic broom, we cut a portion off and found that the bristles were much thicker and tougher at the base. This was to the point that it would be hard for the robot to push the plunger from an off angle unless it could push really hard, defeating one of the benefits of a brush-type mechanism. We tried spreading the bristles out too, but it was hard to find a place to put the brush and even if we could, the tough bristles were very inconsistent. We then tried using a sponge as some other teams also used sponges. However, the disadvantage here was that the sponge did not keep its shape well enough to revert back after each push. It would scrunch up and remain scrunched up. It also had to be cut up to have square protrusions, else it would also have difficulty pushing the plunger at an angle. What we finally settled on was a 3” painter’s brush.

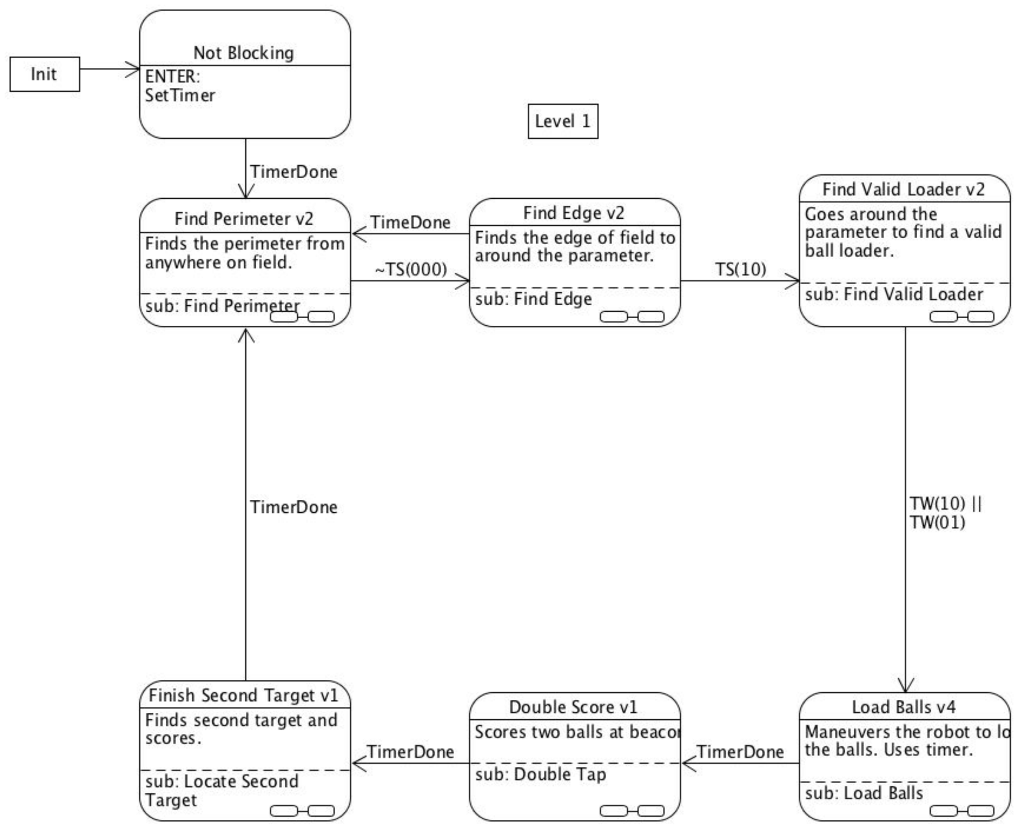
One really good thing about painting brushes is that they are relatively resilient and are really soft. Pushing against the bristles, as long as it is not head on, causes the bristles to spread out. This means that setting the brush at a slight angle is enough to be able to push the plunger even if the robot were off a little. The bristles also go back in place when exiting the plunger, so the mechanism did not need to be reset in between runs.

Though it worked well, there was some problems finding a spot to put the brush. The majority of the robot had already been built which meant that not a lot of available space was left. The boards and most of the circuits already had their spots laid out and a bunch of power wires ran out the back side to the top level. Luckily, the MDF used to support our funnel was cut out with a bunch of holes.

We happened to have exactly two bolts with two accompanying nuts that fit just right into those holes. Though offset and requiring a bit of hammering to get the brush at an angle, we had two holes drilled out of brush’s metal base that we then attached to a hinge. The other side of the hinge was attached onto the holes upside down, so that the brush would be stuck at an angle. Though offset to one side by virtue of the holes being odd across the back, this worked well enough to load balls almost all the time. If we were to do this again, we would probably either space the holes differently or use a different sized brush.

Software Development:

a) Top Level



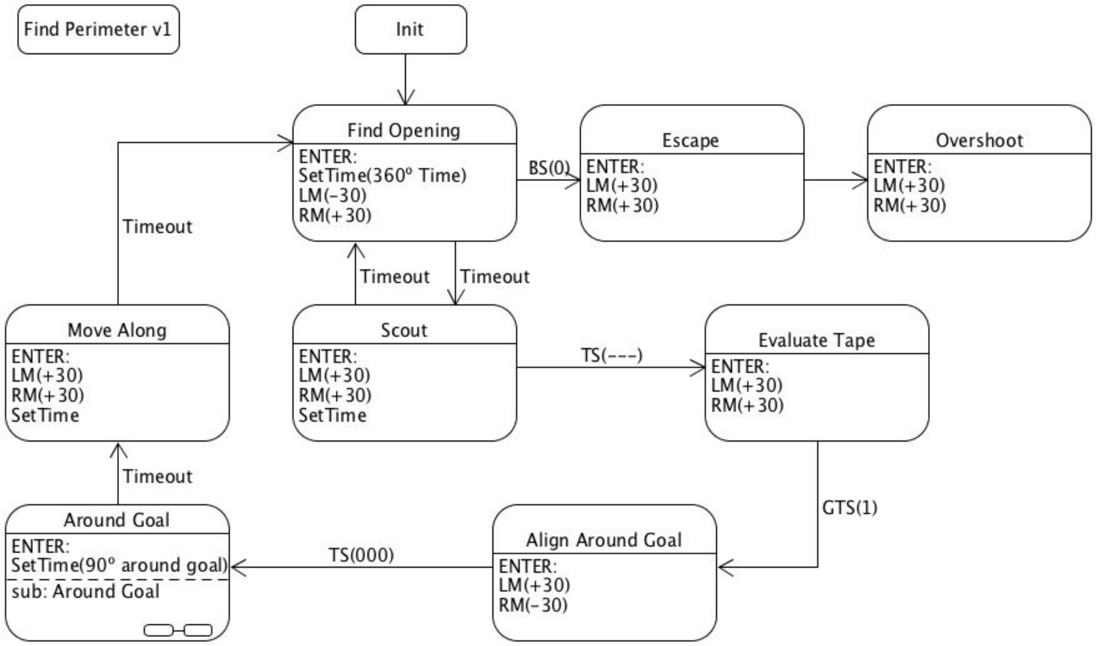
There is a state which acts like a buffer when the robot gets turned on and all the sensors gives false readings which would skip states.

b) States

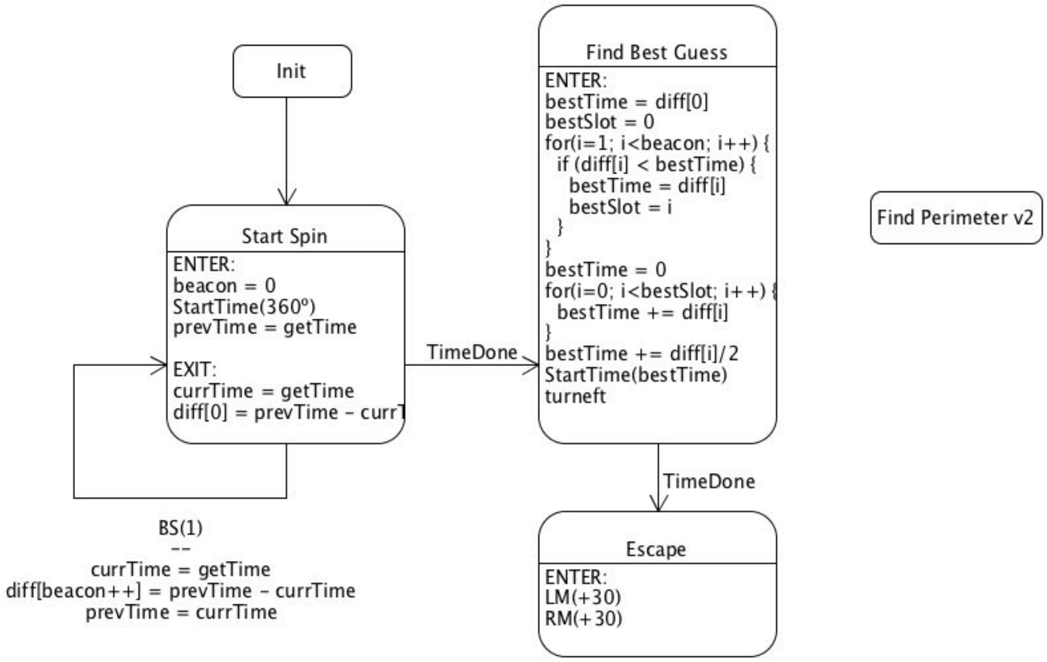
The next step down after the top level design are the individual states.

i) Find Perimeter State

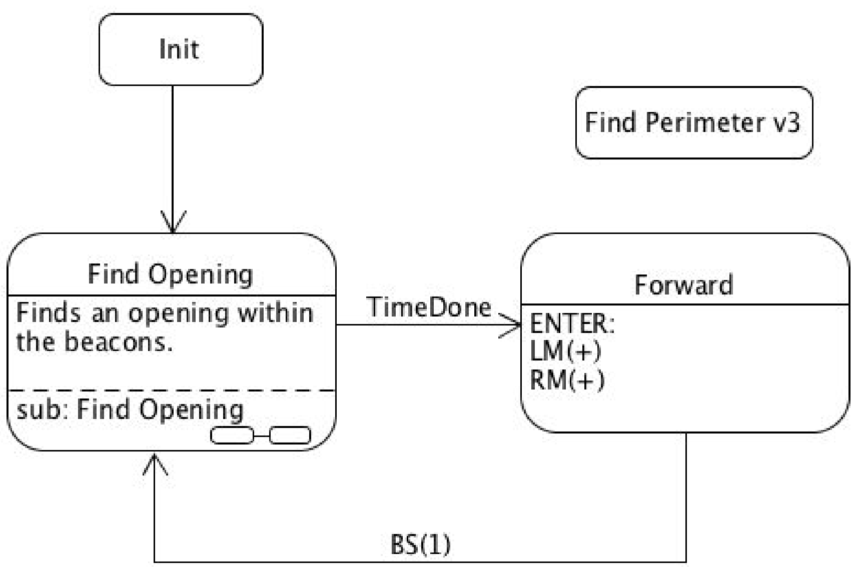
The Find Perimeter State is meant to find the tape on the perimeter from anywhere on the field. The state went through three versions.



The first iteration was supposed complete a full spin and find a location where a beacon was not seen. If the event was not tripped, it would continue forward and attempt to find a location where the beacon was not on again. The trouble with this approach was clipping sides of the beacon if the robot went ahead without a delay stage.



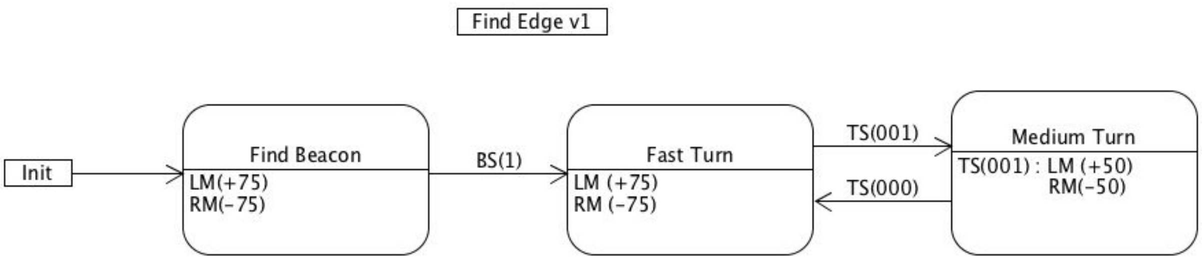
The second version also did a 360º spin and take down the time intervals between each beacon seen. From the beacons, it would return to the place with the largest time interval and attempt to find the perimeter tape. The issue with this attempt is that it took too long. It would take at the very least a full spin at best case and at worst case it would take two spins.



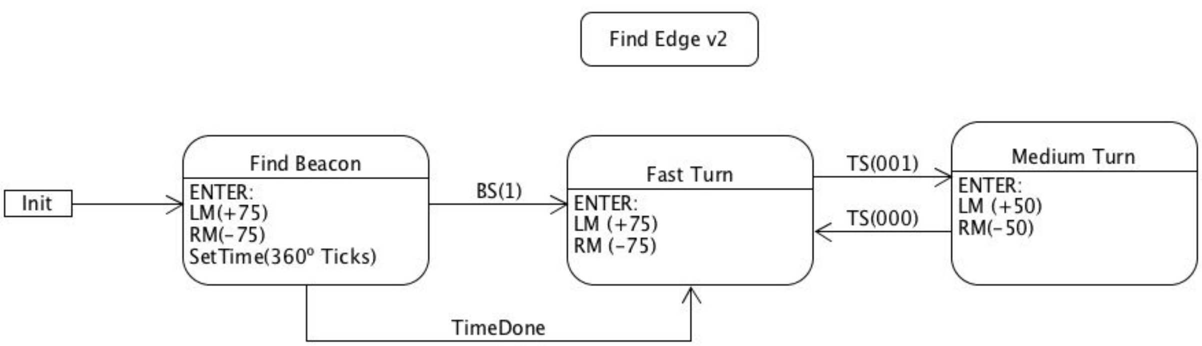
The third version would find an opening within the beacons and drive forward. If it sensed another beacon, it would attempt to find a new opening. This solution cut down on time spent looking for a opening and made sure the beacons were out of the way.

ii) Find Edge State

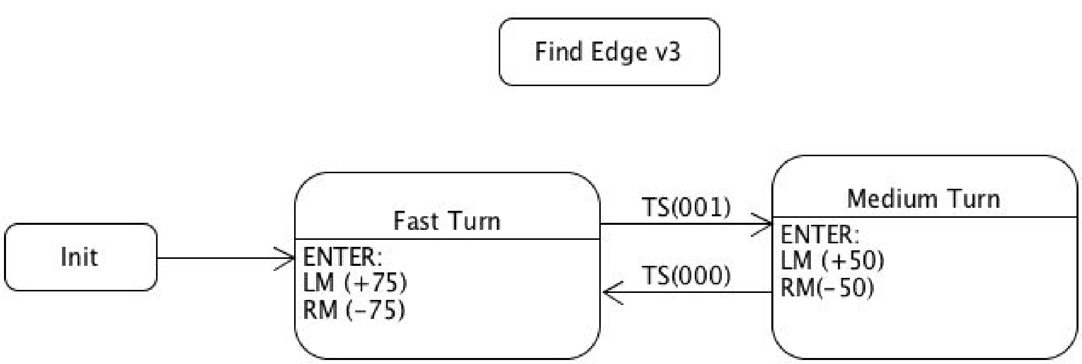
The Find Edge State assumes the robot is placed in any orientation on the perimeter tape and will orient itself onto the tape.



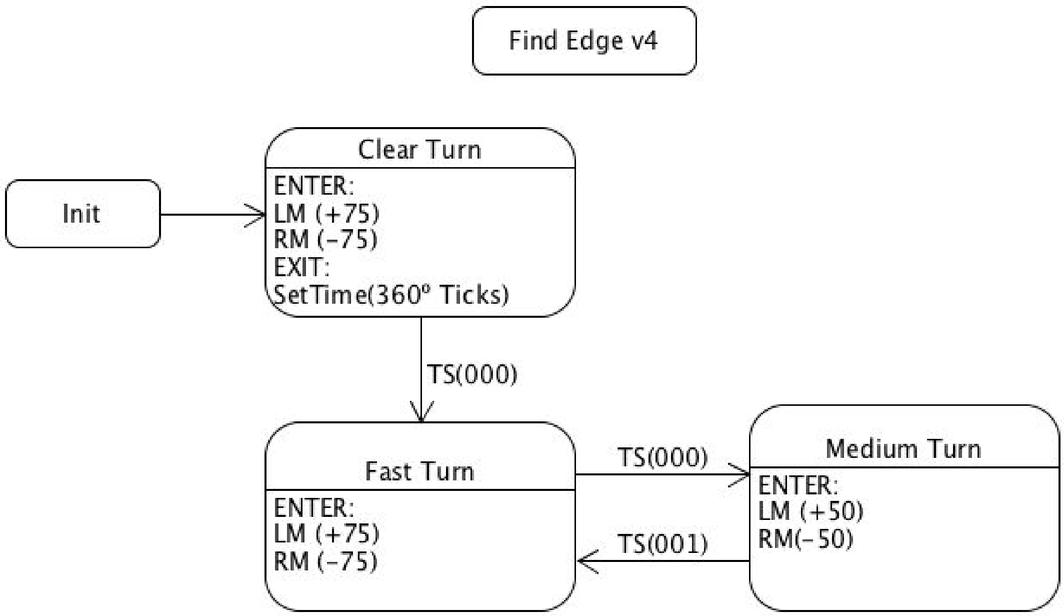
In the development stages the robot would perform successfully if it went counterclockwise around the field. The robot would spin and use the beacon to determine the location of the field and follow the next tape sensed.



The second version of the find edge added in a timer just in case a beacon was not able to be seen.



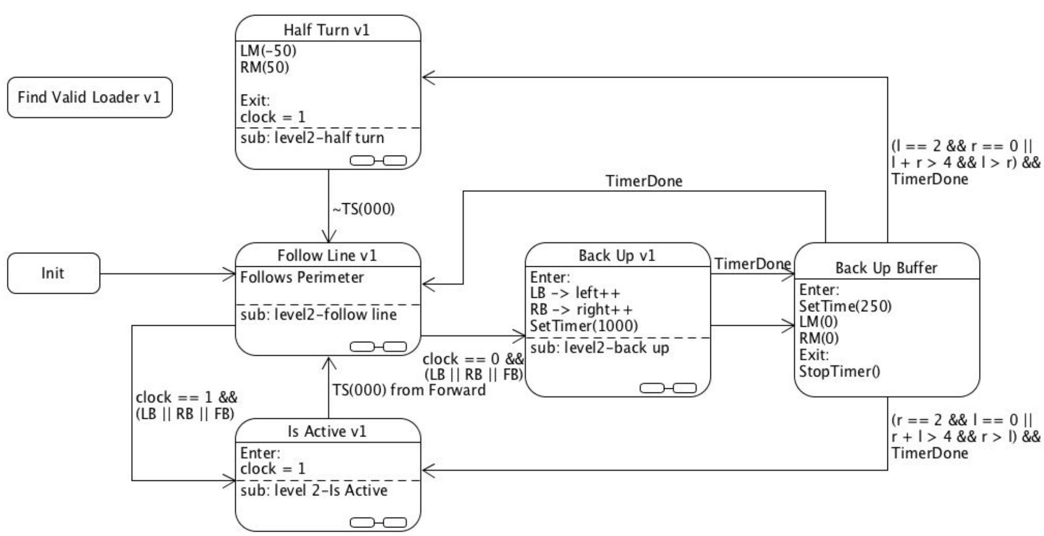
By the time the third iteration happened, the robot would be able to go around the field in any direction so the state which would search for the beacon was removed.



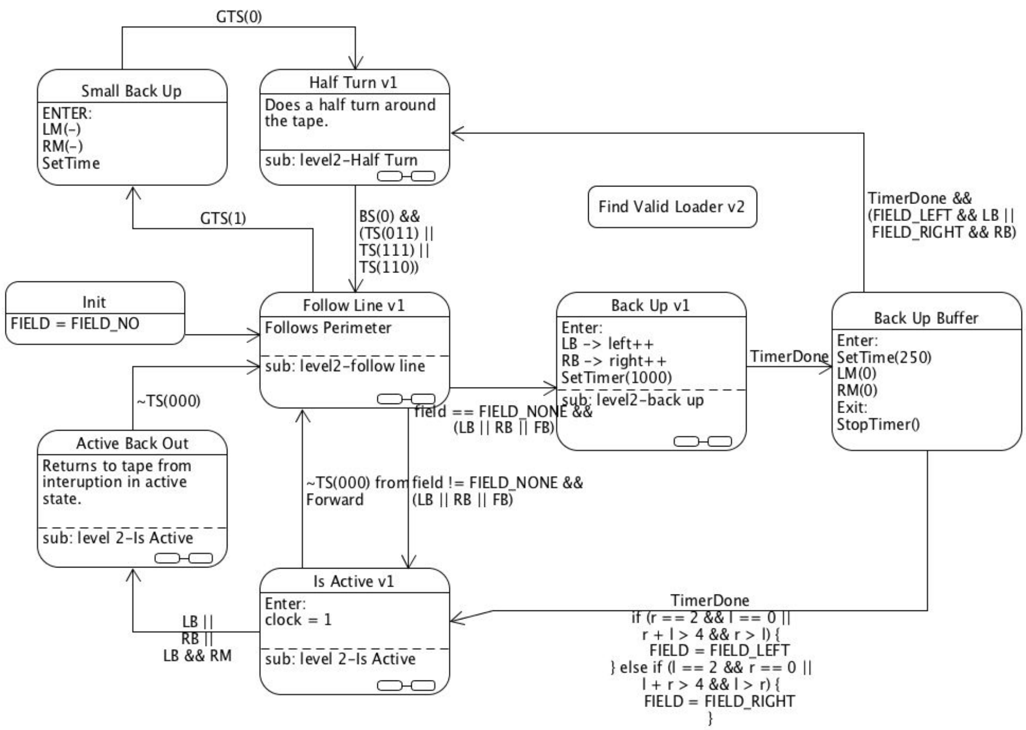
The fourth version accounted for if the robot was perpendicular on the perimeter tape. It would clear the tape and then follow the tape it found.

iii) Is Valid Loader State

The Is Valid Loader State would go around the perimeter and find a valid loader to load balls.



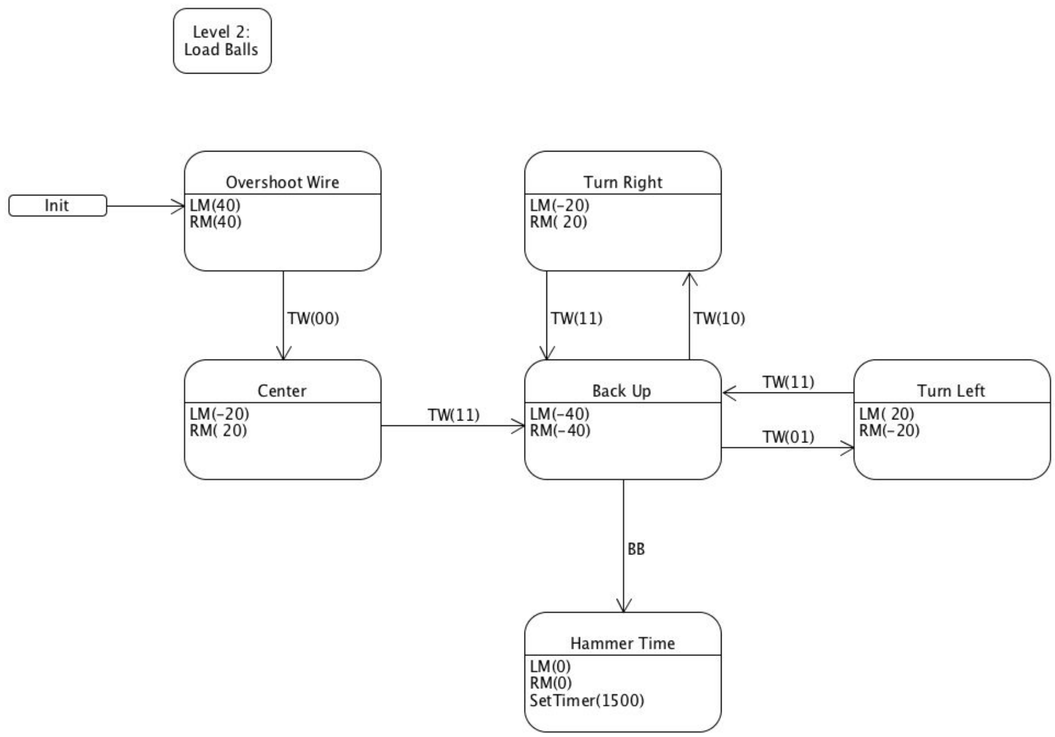
The first iteration could only go counterclockwise. To make sure of this, if the robot determined the tower was on the left side, it would turn around on the tape. A buffer was set between backing up and the next step to prevent the robot from jumping on its back wheels.



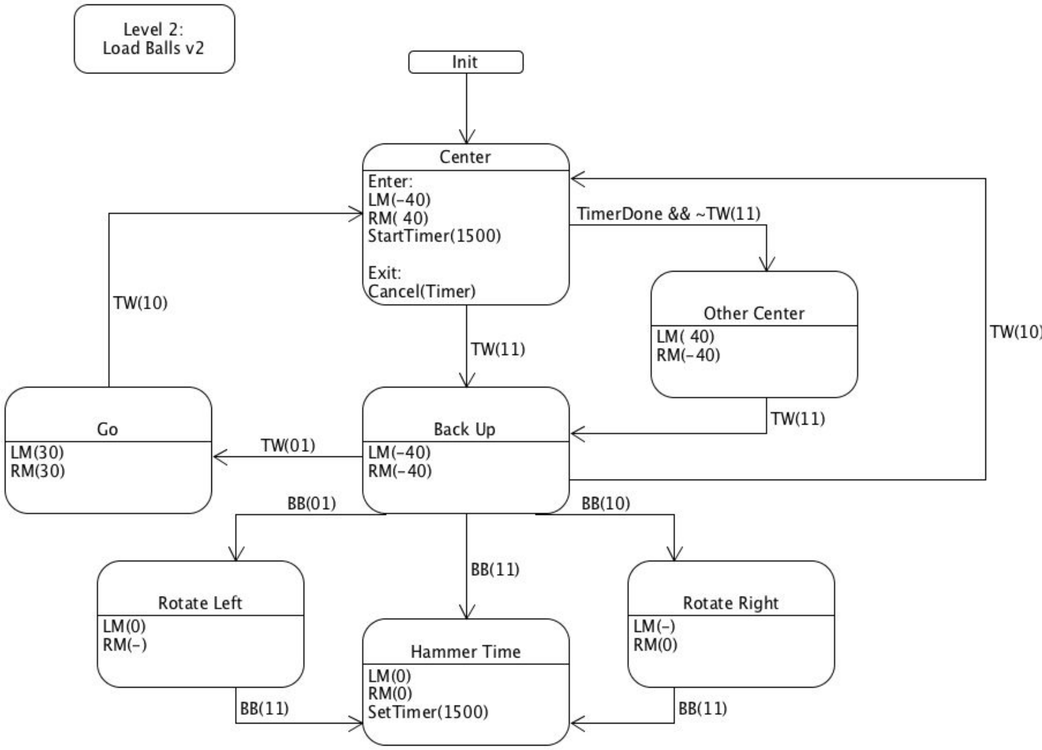
In the second version, the robot was able to go around both clockwise and counterclockwise. The side the field was in relation to the robot was determined from the previous state and taken to account when bumpers were triggered. If a beacon was put on the edge of the perimeter, the robot would either move it out of the way or mount it. If the tape sensor not used for tape following sensed open air, the robot would reverse and make a half turn.

iv) Load Balls State

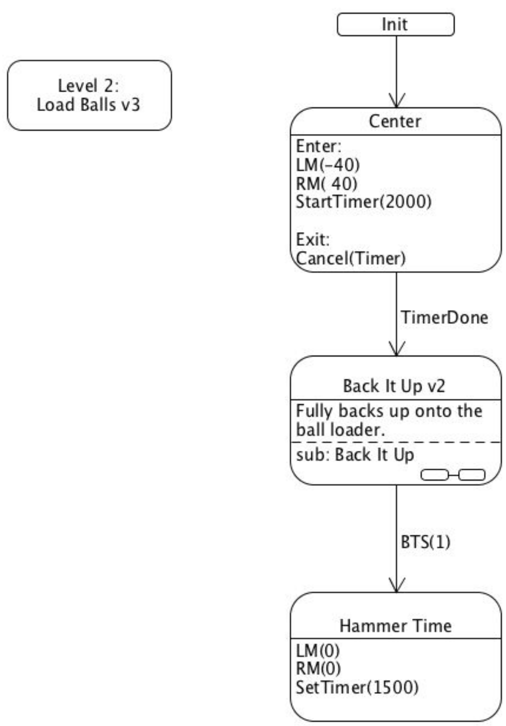
Once a valid tower was found, the robot would load balls.



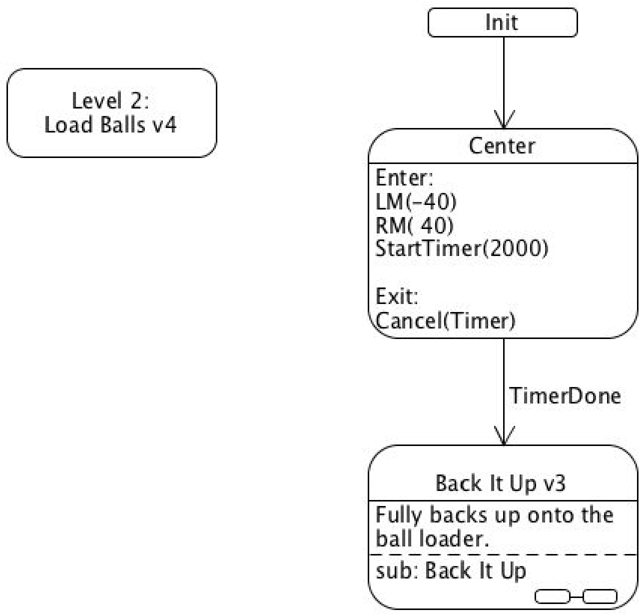
The first iteration would attempt to align itself using two tape sensors. This was not successful because the track wire would have varying widths. The robot would continue trying to align itself in very short bursts by moving back in forth in place with very little progress.



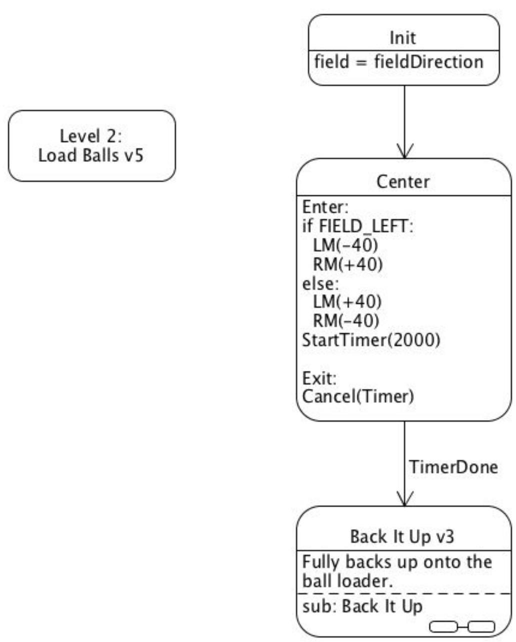
The second version no longer relies on the track wire to align itself or relies on the initial rotation. It uses the rotation to get in the ballpark and uses the bumpers to adjust itself in the proper direction.



The third state factors out the bumper alignment into a state called Back It Up State. The problem with the state is that it would not wait long enough for the balls to fall before adjusting itself again.



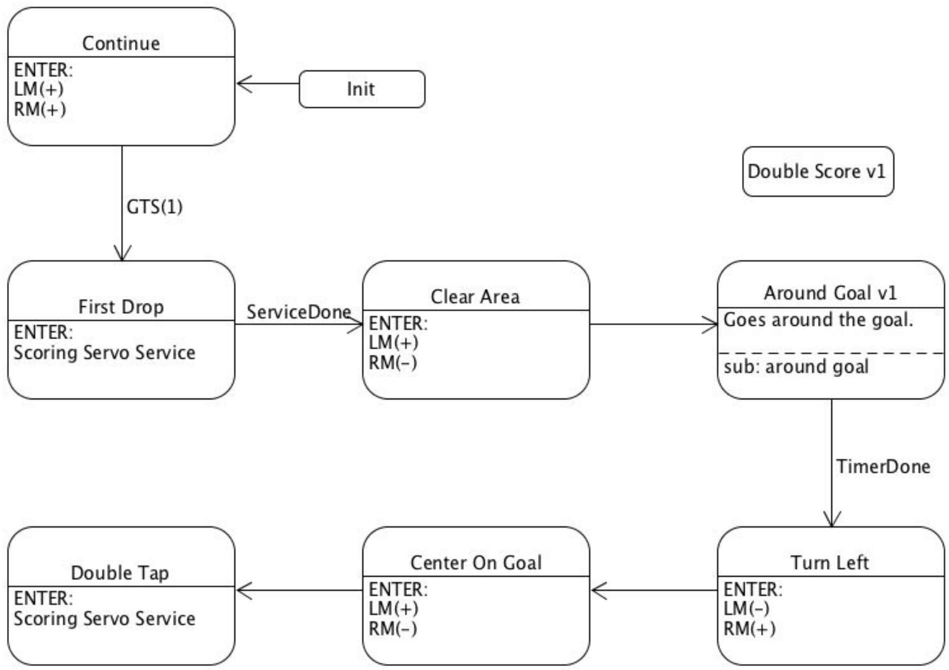
The fourth iteration solves the issue with version 3 by moving Hammer Time state into Back It Up State so the wait a bit before adjusting itself again.



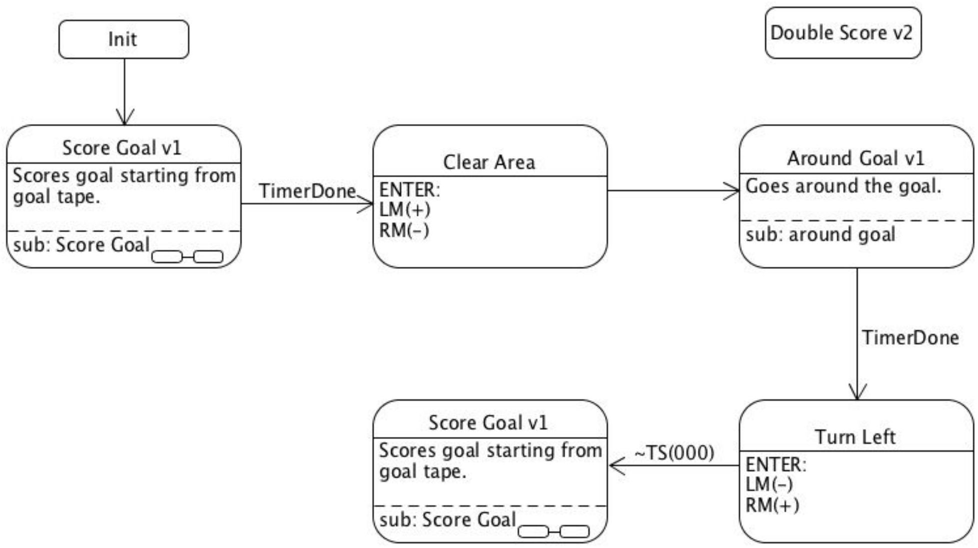
The fifth iteration takes in an argument which allows it to center itself in the correct direction allowing the robot to go around the perimeter in the clockwise and counterclockwise directions.

v) Double Score State

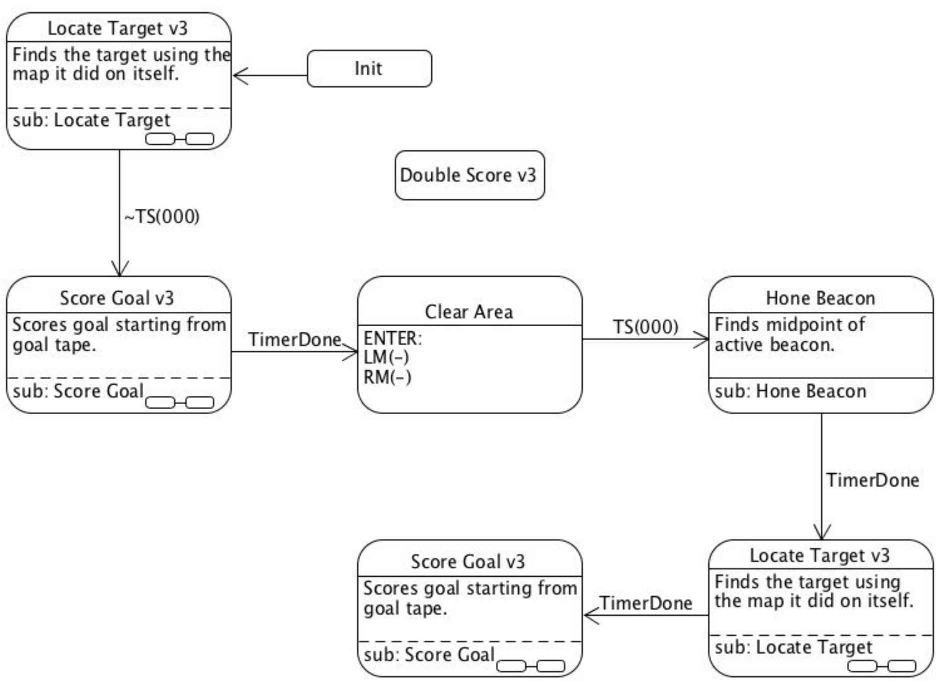
The Double Score State attempts to find a beacon and score two balls at it.



In the first version, after scoring the first time, it would try to tape follow around the beacon. The problem with this is the location of the tape sensors did not make it simple to tape follow around the beacon.



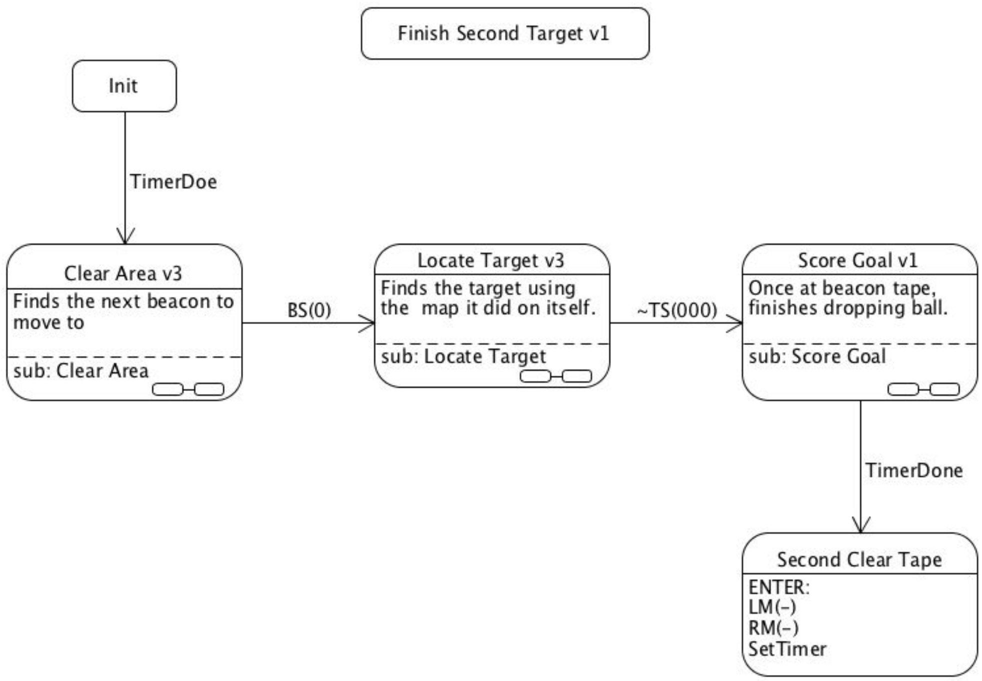
The second version still tried to tape follow around the goal. By this point, the Score Goal Helper State was created. Rather than having two states which did the same function, the state was reused.



During the third iteration, the Locate Target State was changed from being a sub state into a helper state. Rather than going around the beacon, the robot is now trying to score in the same area. But before the second score, the robot readjust itself just in case the first adjustment was not the best.

vi) Finish Second Score State

Finish Second Score State removes the first beacon from its vision and finds a second beacon to score on.



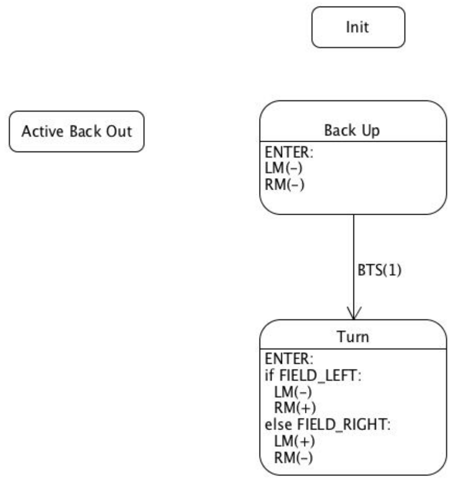
The second target takes advantage of the Locate Target and Score Goal Helper States.

c) Helper States

Helper States are states which perform smaller tasks. It was helpful to remove them from the larger systems for debugging purposes, design small tasks and put them together, and the helper states can be reused. The Helper States are similar to functions.

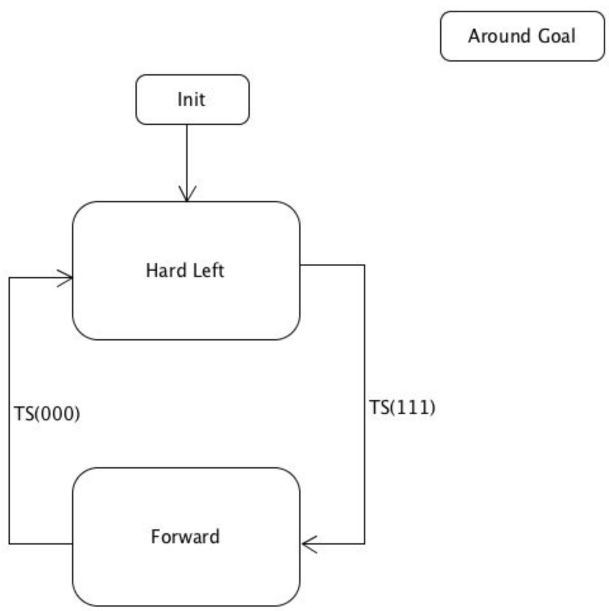
i) Active Back Out State

Active Back Out State is used when trying to see if a track wire is active and is interrupted. It returns itself to the perimeter.

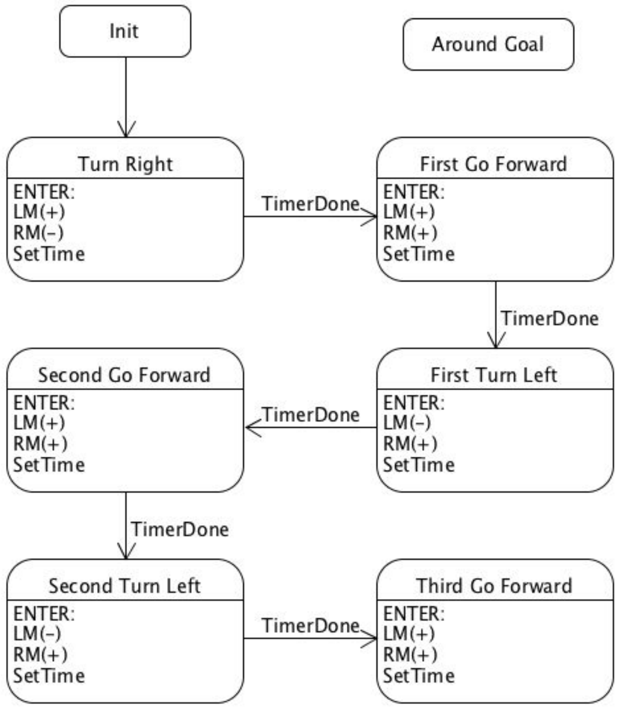


ii) Around Goal State

Around Goal States tries to move the robot around a beacon in search for a second beacon to score on.



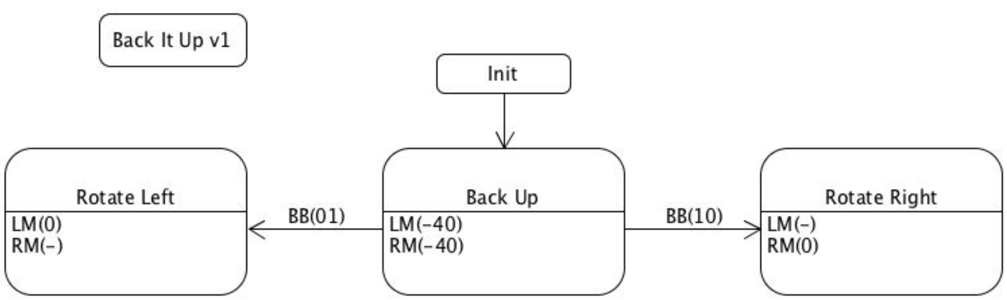
The first version tries to use the front tape sensors to go around the beacon. This approach is less effective due to our tape sensors being very center on the front.



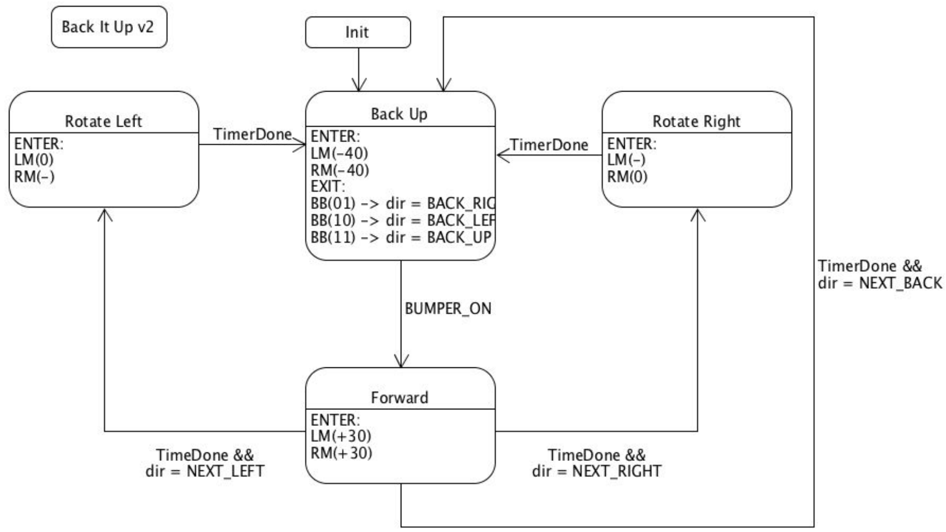
The second version uses timers. The robot surveys the surrounding area and then moves around the beacon.

iii) Back It Up State

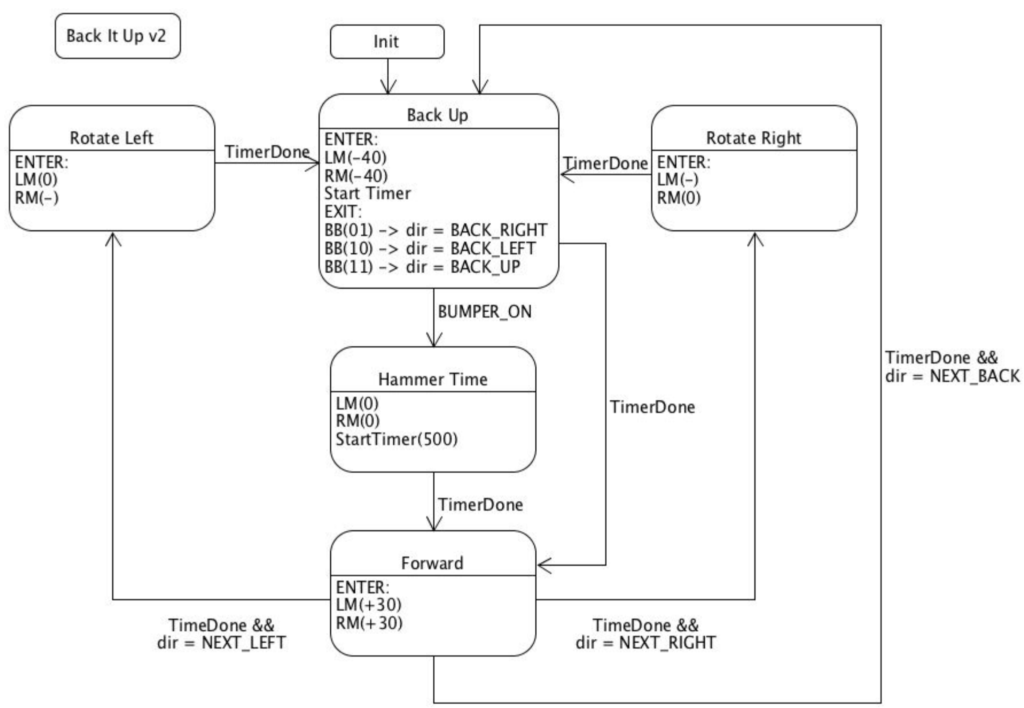
Back It Up State is used to align the robot with the tower.



The first version only takes bumper events into account and tries to rotates in place. The problem with rotating in place is the brush prevents from pressing the bumper another time and it does not have enough force to trip the tower.



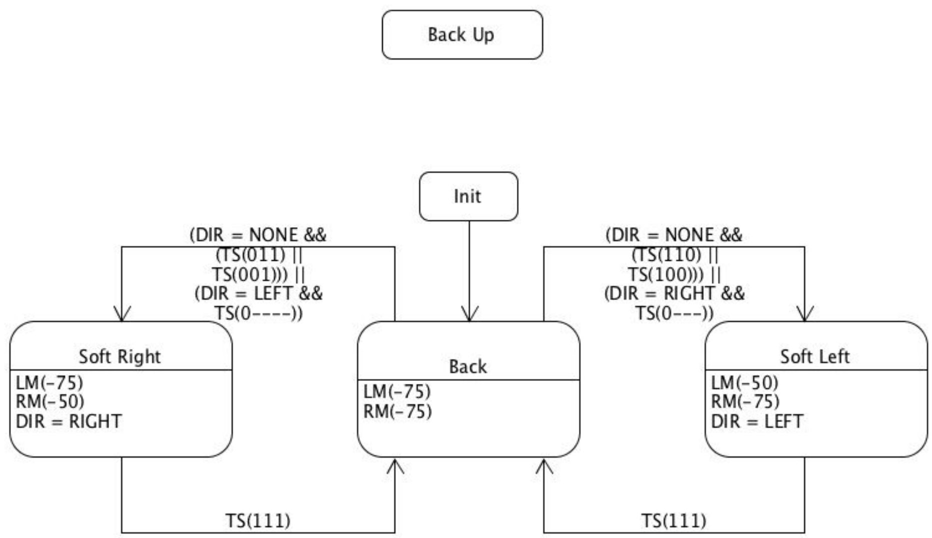
The second version fixes the problems with the previous version by moving forward between attempts.



The third version has a state to wait if balls did get deposited.

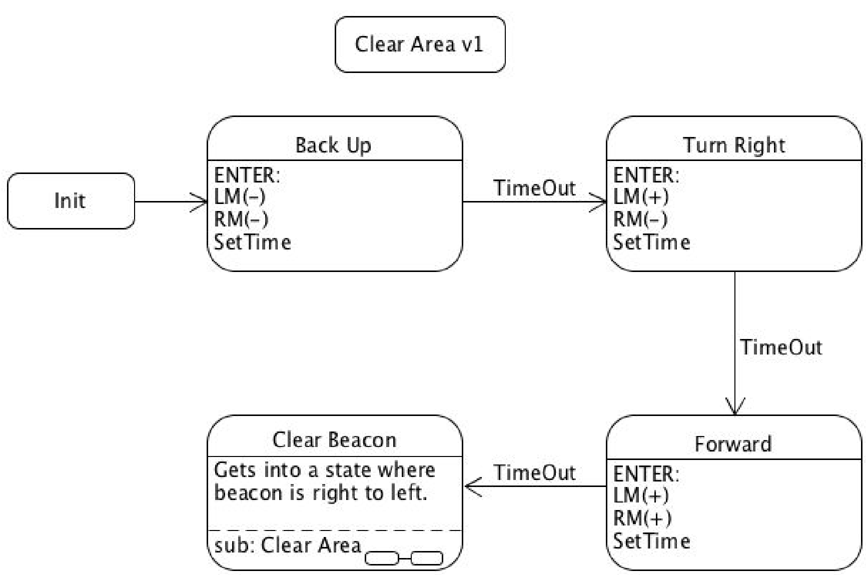
iv) Back Up State

Back Up State tape tracks backward.

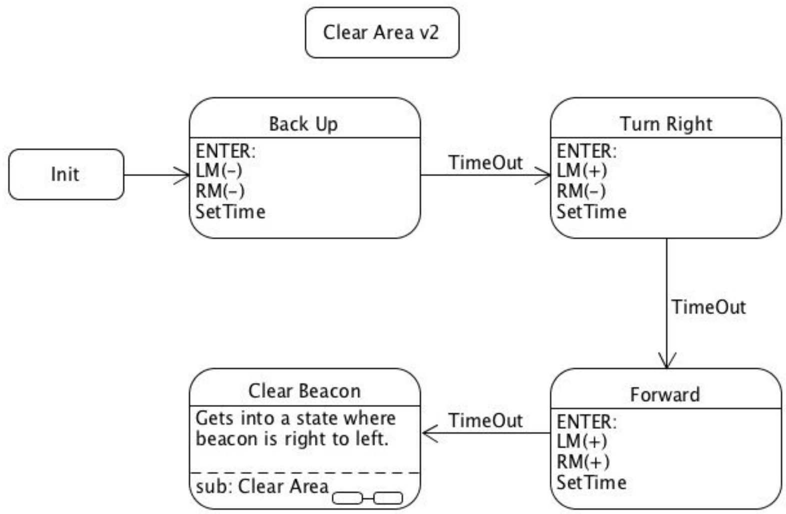


v) Clear Area State

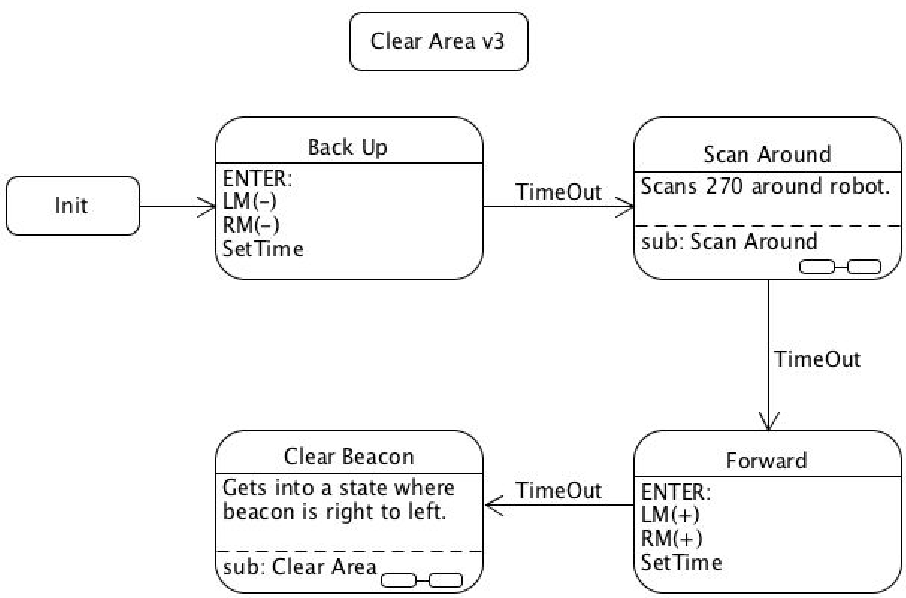
Clear Area State removes the robot from the surrounding area.



Note: The first version of Clear Area was lost.

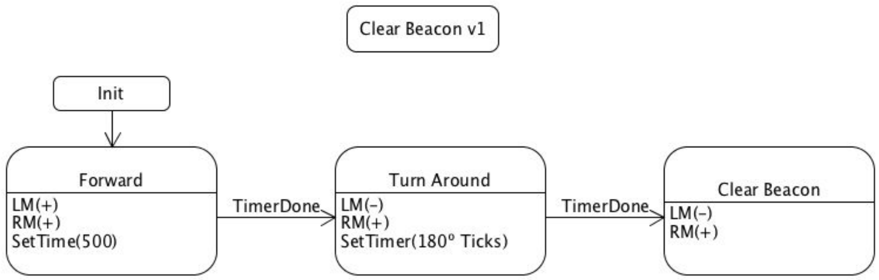


The problem with the second version was it did not take into account its surroundings while looking for the next beacon.

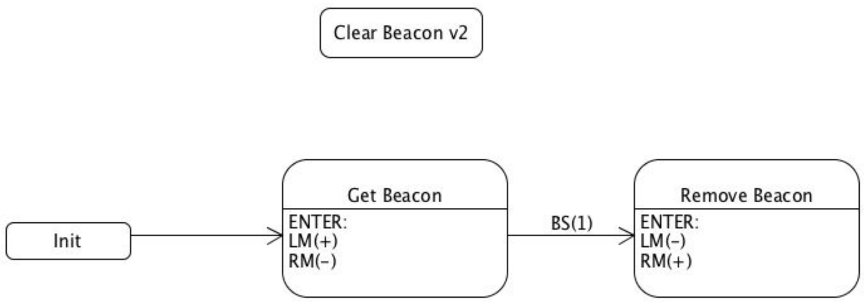


vi) Clear Beacon State

Clear Beacon State removes the beacon from the robot's vision.



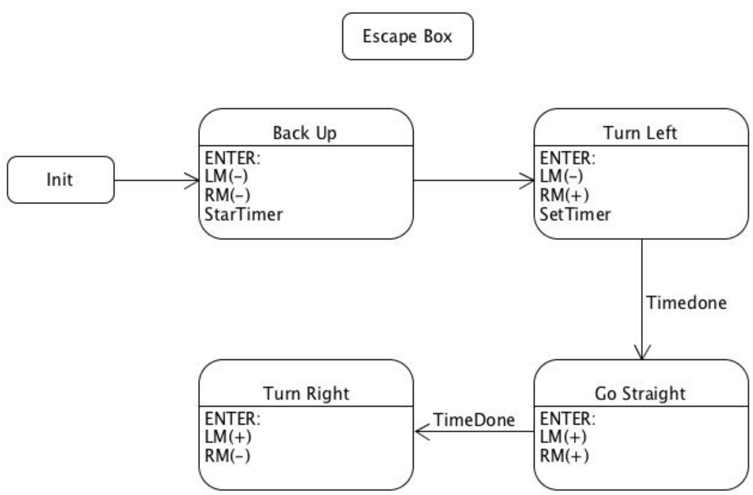
The first version was meant to be part of a specific state.



The second version was made to be more general.

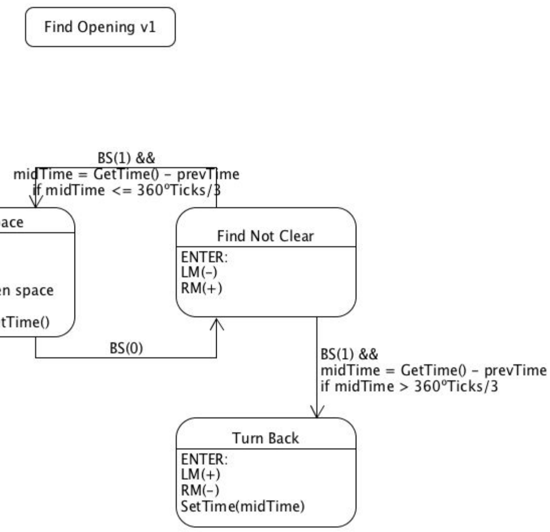
vii) Escape Box State

Escape Box State was used to remove the robot from a situation where there was a blocking force infront and behind which normal collision would not account for.



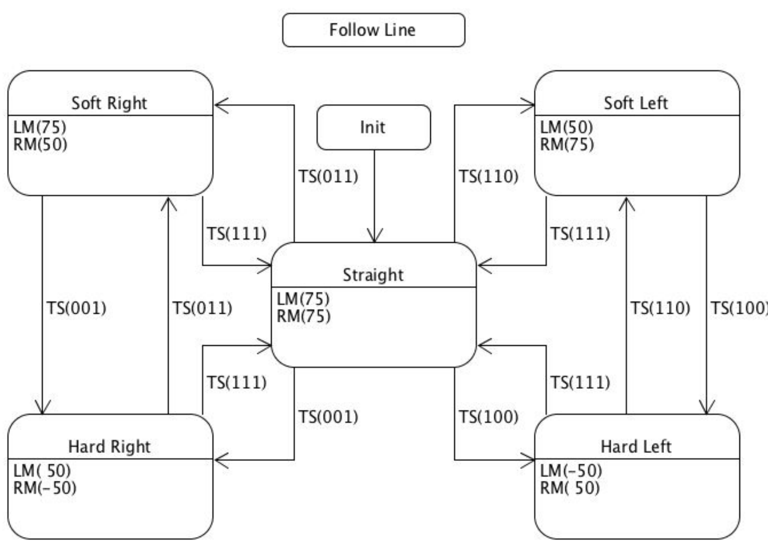
viii) Find Opening State

Find Opening State found between beacons for the robot to move through.



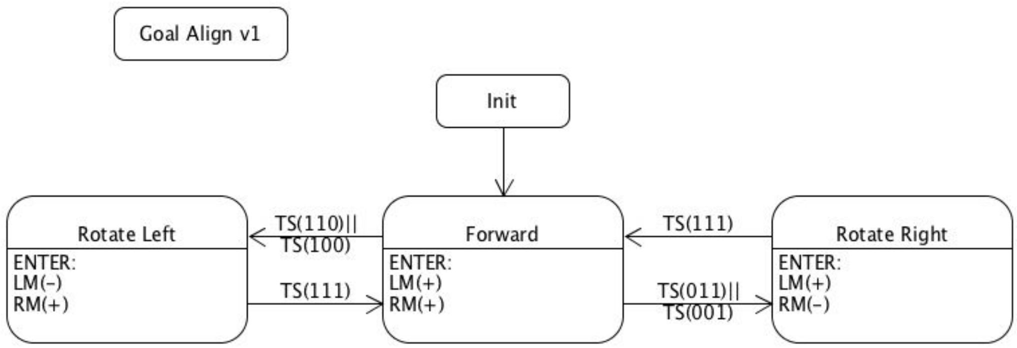
ix) Follow Line State

Follow Line State tape follows.



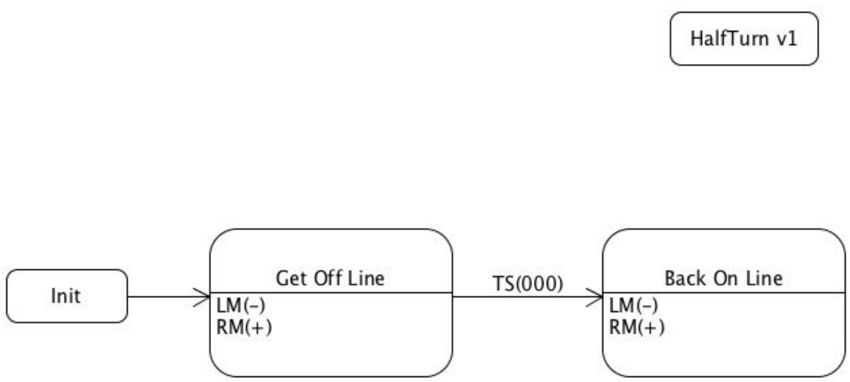
x) Goal Align State

Goal Align State helps the robot align with the goal once it has found and driven up to the beacon.

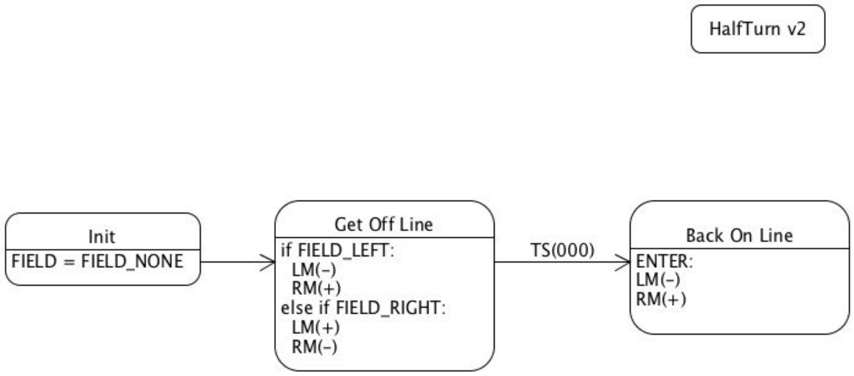


xi) Half Turn State

Half Turn State turns the robot around on the perimeter tape.



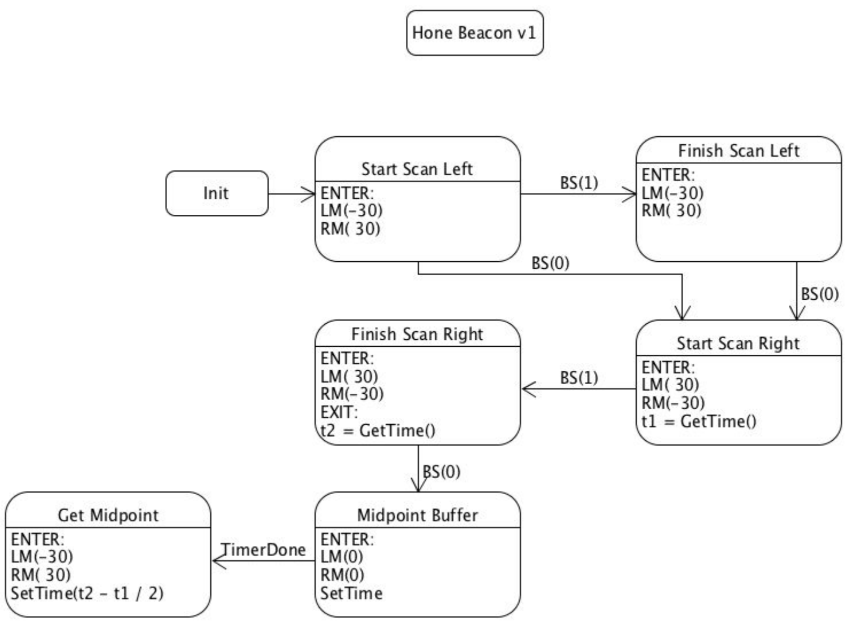
The issue with version one is depending on which side the field is on, turning might make the robot fall off.



The second version takes into account which side the field is on to make sure the turn is clean.

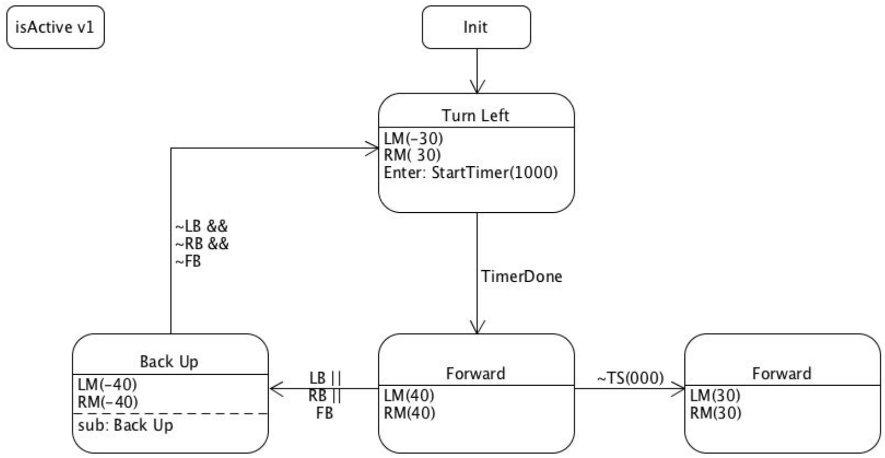
xii) Hone Beacon State

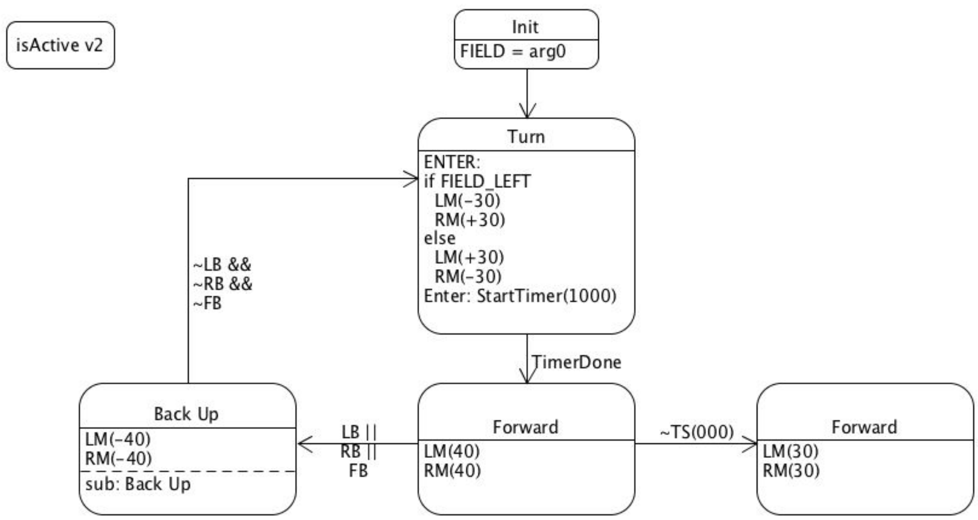
Hone Beacon finds a beacon and centers itself to the beacon.



xiii) Is Active State

Is Active State dashes across the tower to check if it is valid.

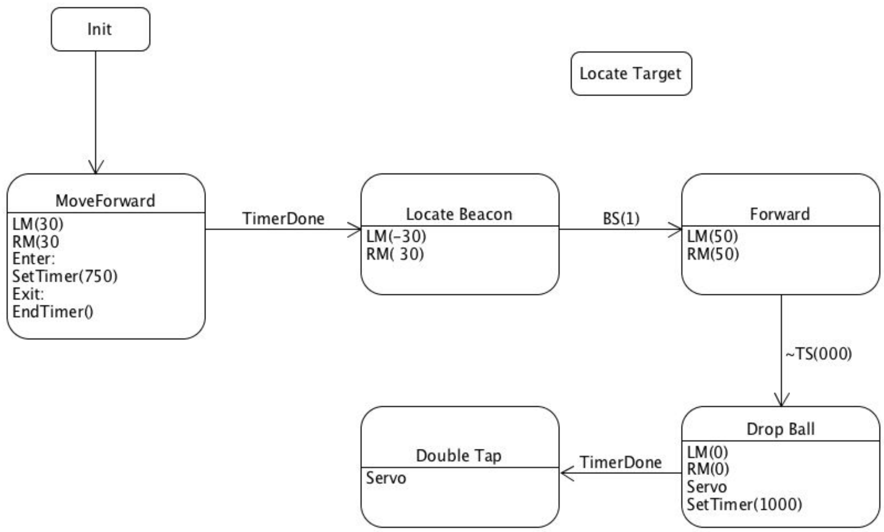




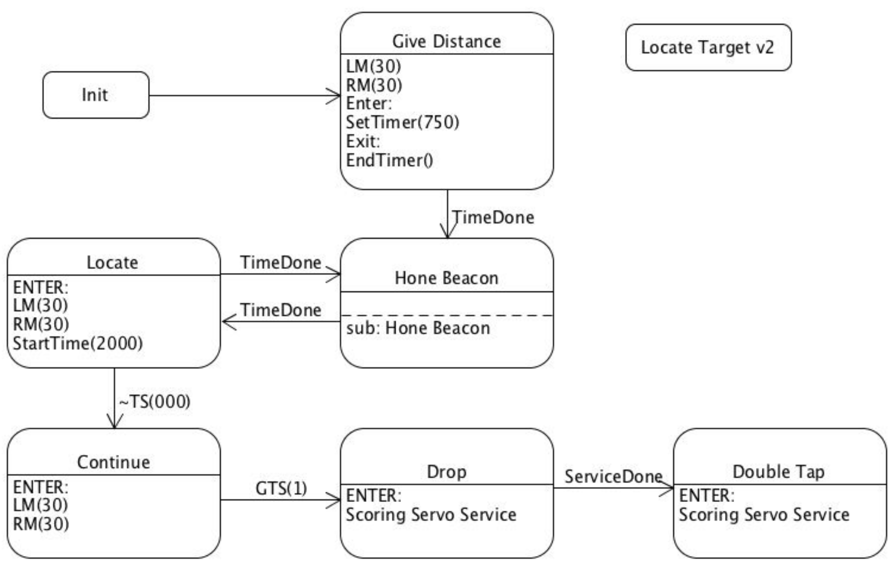
The second version enables the robot to go around the perimeter tape in both the clockwise and counterclockwise directions.

xiv) Locate Target State

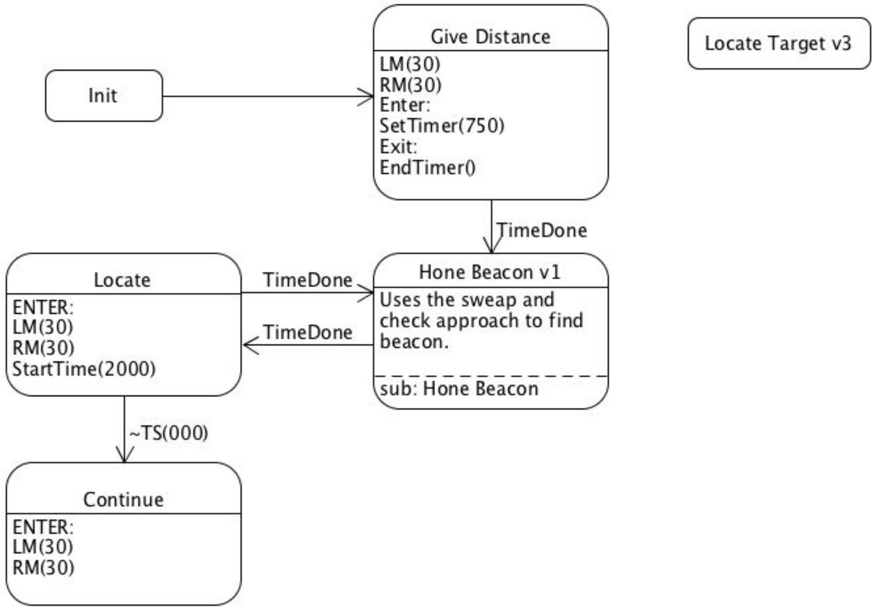
Locate Target State finds a beacon and travels until it reaches the tape.



The first version only checks once and it would lead the robot to the side of the beacon rather than to the beacons center.



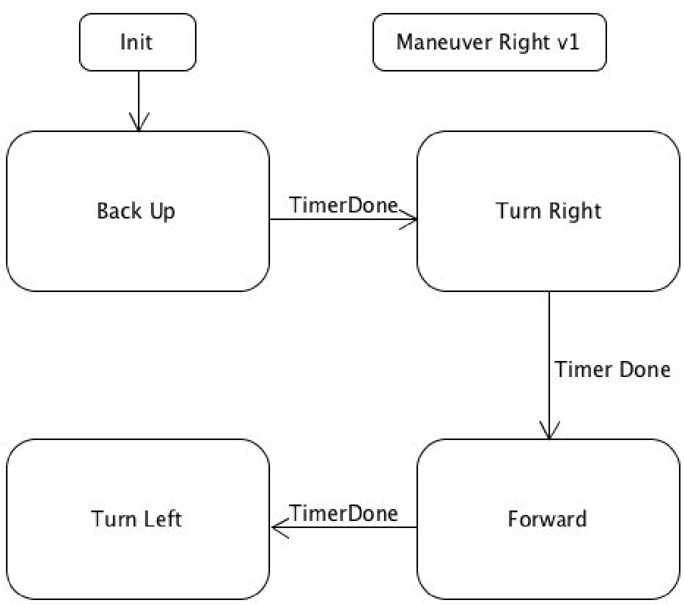
The second version uses the Hone Beacon Helper State.



The third version is the transition from Locate Target being a Sub State into a Helper State.

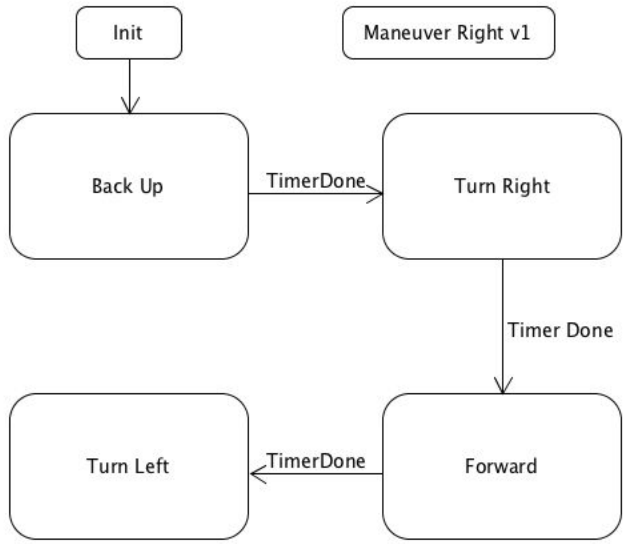
xv) Maneuver Left State

Maneuver Left State is the collision resolution if the robot gets hit on the left side.



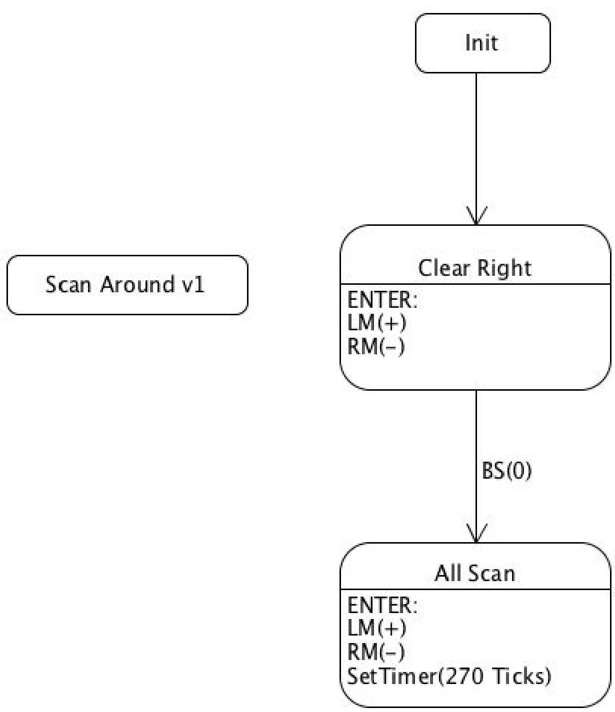
xvi) Maneuver Right State

Maneuver Right State is the collision resolution if the robot gets hit on the right side.



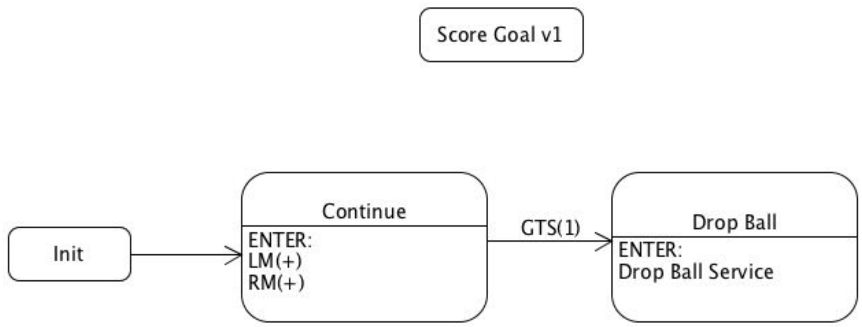
xvii) Scan Around State

Scan Around State scans the area around the robot.

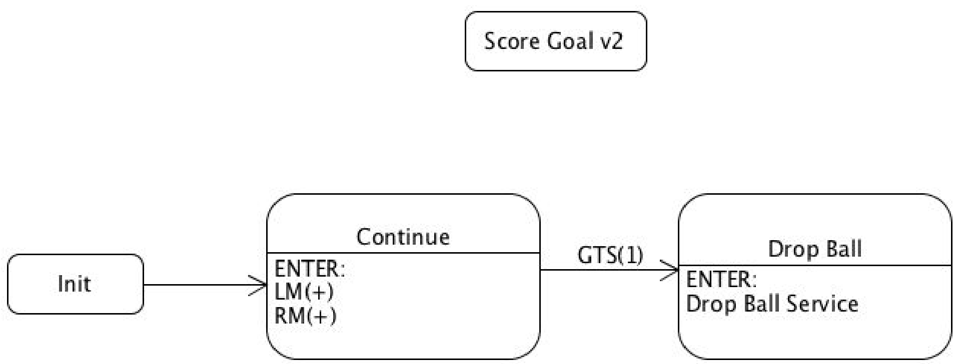


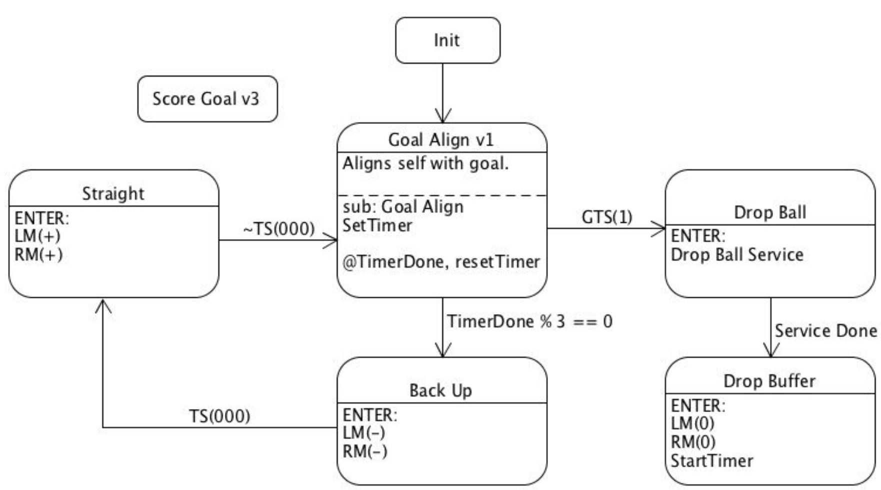
xviii) Score Goal State

Score Goal State aligns itself with the beacon and scores.



Note: The first version of score goal was lost due.





The third version samples the goal tape sensor to check if it has already been aligned. If the robot has not aligned well enough within a time period, it reverses and tries to align itself again.

d) Containers

Find Perimeter State and Locate Target State were put into containers. Containers allow a state to react to bump or tape events and give them priority over other events. With these two states, after a collision resolution or tape event, the robot can not return to the state so it would have to reset each time. By having states in containers, there is no need to modify the core state and it can be act as an overarching behavior of a state which can be modified in one place.

Integration of Individual Elements:

a) Electromechanical Integration

After all the components had been laser cut the sensors were placed into the positions that were assigned in the Solidworks model. This was arguably the easiest part of the project since all the work for the electrical and mechanical had been developed in conjunction, making the incorporation phase painless. All the holes lined up in the correct location and the boards were secured using standoffs and #4-40 screws in ¼" and ⅜" varieties. The only difficult portions of this integration were wire routing since a large amount of connections needed to be made and the requisite wiring was intensive, and comparator values were not stable enough to use with non-ADC pins. Use of the hinges made positioning routing much easier but as additional wires were added for modules that were not initially determined to be needed the wiring began to look messy. While this would have been reasonably easy to remedy there was not sufficient time to deconstruct the robot and rewire everything. Even with the temptation of being able to paint the robot any color we wanted there was no motivation to deconstruct the robot.

b) Software/Electrical Integration

i) Actuators (DC Motors, Servo)

The PWM library was used to interface with the DC motors.

The servo was interacted through the RC library provided. According to the datasheet, the minimum pulse time was 1000 for -90º and the max pulse time was 2000 for +90º.

ii) Sensors (Bumpers, Tape, Track, Beacon)

The bumpers were read in through using the IO_Ports library.

The tape sensors were read through using the AD.h library. The function accepted whether the LED is on or off during the readings. Analog values would be better used as analog values to be able to compare against thresholds or for synchronous sampling.

The track wire sensors were read in through the AD library. The high output of the track wire was too low to be detected by the IO_Ports library. To resolve the issue, the value from the track wire sensor was read through the AD library and compared to a value to determine whether a track wire was sensed.

The Beacon was read in through the AD library. The values outputted by the beacon detector were not high enough above the threshold to be considered 1. To solve this issue, an analog value was read and determined whether a beacon was being sensed or not.

iii) Drivers

The drivers were written and tested within a day.

iv) Harness

A test harness was written in the spider.c file. There are two options when running the harness. Option one allows testing of individual components which interacts with the drivers and displays the values read in. Option two tests all the sensors together using event checkers.

v) Music App Development

To play music through the speakers, a mobile app was created. The mobile app was written in React Native and compiled into a Android App. A website was hosted on Heroku using an Express, a framework for Node.js, server. Firebase was used to take advantage of it's real time updating capability. When the song choice was changed on the website, firebase would send the new values to the mobile app. The mobile app would use this value to know which song to play. The app currently only plays songs that have been uploaded to it. An improvement to be made for the app would be to integrate either the Spotify mobile API or the React Native YouTube library to allow songs not downloaded to be played.

c) Bill of materials

The complete list of the materials needed for this project is below and comes in just below the $150 price point limit set by Professor Elkhaim.

Materials		Number	Cost Each	Total Unit cost		Total Cost

Wheels		2	$4	$8.00		$141.47
Casters		2	$2	$4.00		
Standoffs (male/female)		10	$0.60	$6.00		
Standoffs (male/male)		8	$0.45	$3.60		
MDF sheet		4	$3.50	$14.00		
Motors		2	$13.64	$27.28		
Perboard (Small)		1	$0.60	$0.60		
Perfboard (Medium)		2	$0.85	$1.70		
DC Motor Couplers		2	$6.93	$13.86		
M3 Screws		4	$0.17	$0.68		
4-40 Screws		22	$0.17	$3.74		
RC Servo		1	$3.24	$3.24		
Limit Switch		3	$1.12	$3.36		
5V LDO		1	$1.00	$1.00		
3V LDO		1	$1.25	$1.25		
Female Molex		5	$6.00	$30.00		
Male/Female Jumpers		50	$0.12	$6.00		
ABS Pipe		1	$1.70	$1.70		
45 degree elbow		1	$2.79	$2.79		
Hinge (4 pack)		1	$4.99	$4.99		
10-32 screws 4"		2	$0.53	$1.06		
10-32 screws 1 1/4"		2	$0.43	$0.86		
10-32 Nuts		8	$0.09	$0.72		
10-32 Lock Nuts		8	$0.13	$1.04		

Journal Entries

Progress Report 11/4/2016

Josh Passmore:
Designed Trackwire circuit on protoboard on 11/3/2016
-> perf boarded two track wire circuits 11/4 on one protoboard -> 100% complete and functional.
-Bels out of regulator L00s
-Testing of two beacon sensors needed.
-Tape sensor prototyped on protoboard 11/3/2016
-Got check off for 3 designs -> received team kit (#16) unostack, DS3648, ULN2003A, Stepper Driver, H-Bridge

Julia Warner:
Began basic design for solidworks model.
Created and edited main project global variable/equation sheet.

Josh Pena: In Seattle.

Progress Report 11/5/2016

Joshua Passmore: Designed tape sensor circuit on protoboard
->Soldered 4 tape sensor circuits onto on perfboard
-> Sensors completed
->Challenges -> Working alone takes a while and requires extra focus to prevent errors. LD0 worked on protoboard for 3.3V, but I accidentally reversed it on the perfboard-> had to reverse and desolder.

Julia Warner: Continued working on solidworks model.
Got basic design for all parts, though very rough.

Josh Pena: Still in Seattle.

Progress Report 11/6/2016

Joshua Passmore: Touched up circuits, made specialized connectors, verified all functionality.
Created Gantt Chart with relative scheduling to meet beer checkoff.
Worked on Solidworks design -> Found models for wheels, limit switch, ball casters.
-Finished bottom level assembly and mounting configuration.
-Still need to do second level w/ hinge mechanism, ball release/collection, et. -> Need BOM for materials.
Wheels delivered -> Now have wheels, limit switches, motors, and some hardware.

Julia Warner: Not present.

Josh Pena: Not present.

Progress Report 11/7/2016

Joshua Passmore: Made significant progress on solidworks.
Made minor adjustments to circuit connections.
Laser cut MK1 prototype.
Made some mistakes with mechanical holes.

Joshua Pena: Confident he won't blow shit up. Setup test harness for spider.
->Need to finish sensor inputs (ports)
->tape, servo, motors(bidirectional)

Julia Warner: Helped other members.got MDF and foamcore from BELS.
Acted as support for Josh Pena on code.
Provided input on bumper design.

Progress Report 11/8/2016

Joshua Passmore: Retrieved ball caster wheels.
Developed solidworks further.
Lasercut second iteration of secondary level.

Soldered power distribution board.
Created more custom wiring for modules.
Bought standoffs to mount all boards.
Helped oversee sensor integration into software and mechanical structures.

Joshua Pena: Finished up drivers. All but bumpers tested.
Moved state machine from paper to files.
Reviewed service.
Created event checkers.
Tested all but bumpers.
General trouble with track wire sensors.

Julia Warner: Worked on writing a service function for track wire sensor.
Stayed until 8pm in lab. No members communicating with me.
Left early. Submitted available times to Joshua Passmore.

Progress Report 11/9/2016

Joshua Passmore: Soldered one custom power wire.
Worked on solidworks design.
Setbacks: Solidworks crashed and work was lost.
Design for tower was released and current version of ball release was inadequate. -> Massive bot redesign necessary and ongoing.

Joshua Pena: Fixed track wire functions -> went to AD.
Created and tested all events (except bumpers)
Fixed up tape service.
Started state machine.
Finished first substate, working on second one.
State machine not tested yet.

Julia Warner: Worked on getting a wheel modules setup.
Josh (Passmore) asked me to continue working on it.
Finished, still asked to add more helper functions.
Confused on how I'm supposed to contribute.
Attempted to help with prototyping. Still no mention of meeting times.
Submitted available times to Joshua Pena through given website.

Progress Report 11/10/2016

Joshua Passmore: Finished redesign of solidworks model.
Lasercut new robot design.
Went to hardware store and obtained hardware and ABs pipe.
Assembled robot except for wheels, ball collection, and servo.
Soldered new, longer power connectors.
Helped Josh Pena w/ extensive testing of system.

Hooked up all wiring on robot.

Joshua Pena: Finished up coding state machine.
Debugged state machine.
Finalized harness (except bumpers).
Wrote helper functions.
Wrote up potential mapping.

Julia Warner: Finished helper functions.
Josh Passmore continues insisting I work on helper functions.
Still wondering when we are supposed to meet. I have no idea of the other members schedules.
Been checking my phone every 10 minutes for reply. Stayed home and started learning HTML to make website while I wait.

Progress Report 11/11/2016

-> Mandated meeting: Issues with group communication between Julia and rest of team. Issues finding ways to help without being told.

Joshua Passmore: Lasercut wheel hubs -> moving robot now functional.
Soldered new perfboard for bumper actuators w/ indicator LEDs.
Finished assembly of sensors onto robot-> only code and release mechanism remaining.
Helped Joshua Pena w/ code debugging and state machine debugging.
->Setback: a stray letter "x" was accidentally placed in a file (ES_Configure.h) and was unable to be found for 1:20min.
Still working on DC couplers -> may not use.
Soldered and installed remote switch for robot.
Cycled battery once.
Worked out basic configuration of release mechanism.

Joshua Pena: Bumper harness done.
Test harness done.
Bumper events good.
Fixed up tape sensor. No synchronous sampling but mech.

Julia Warner: Worked on code development for services and website HTML.
->Left early to continue working on it.
->Difficulty working on website in lab due to all computers being used.

Progress Report 11/12/2016

Joshua Passmore: Rewired robot to improve cleanliness, efficiency
->Made custom cables and color scheme.
->Documented in notebook.
Integrated bump sensors into design -> screwed into position.
Taped bumper design onto requisite positions.

->Works well - needs slight modifications.
->front bumpers should be extended.
Rear bumpers should be filleted to reduce overhang.
Attachment mechanism needs to be devised.
->Tape works great but is temporary.
Helped Josh Pena w/ code review and state machine ideas/debug.
Cycled battery twice.

Joshua Pena: Had issue with 2 ES_Config- removed '.' From include.
Helped finish up wire and testing for components.
Bumpers were added.
Can (more than less) follow line.
Can attempt to turn around and find track wire.
Rewrote code to include ES_Entry.
Changed location of all substate Init.
Having issue with track wire. Not sure if bumping, movement, or general wires are to blame.
Check track wire to make more solid readings.
Core finishing up (2/4) can start with corner and interferences.

Julia Warner: Showed up at Joshua Passmore's request.
Continues to work on HTML code.
Left early due to issues working on Website (working with large laptop with mouse on my lap).
Team has still not communicated with how I can help. Meetup times still not mentioned.

Progress Report 11/13/2016

Tutor meeting scheduled for Tuesday at 13:20. Said to have design ready, but we're well past that. Sent email to Gabe regarding team issues.

Joshua Passmore: Machined PVC to allow ball collection.
Designed and cut new MDF ball loading funnel.
Diagnosed problem w/ tape sensor not reading -> IR LED separated from mechanical connector/shroud and was not visible to phototransistor.
->Loosened connections, added electrical tape.
Diagnosed issues with trackwire solenoid position -> relocated. solenoid to determine coordinate more precisely.
Helped Joshua Pena with troubleshooting robot.

Joshua Pena: Update backup algorithm, use Tape Sensor 4.
Updated code to involve tape sensor 4.
Changed load balls algorithm twice:
1) Use two tape sensors

2) use timer

Can successfully load balls.

Added buffer when switching directions to avoid robot from doing wheelie.

Tape sensors still challenges, were taped up again. Used android phone app to check if it functions.

Tried to use android phone to debug but was unable to connect.

Julia Warner: Came to lab when Josh Passmore texted me again.

Showed what I've done on website.

Still no mention of meeting times. Almost done on basic website design.

Left due to teammates ignoring me. Got website mostly done.

Fixed issue with text boxes preventing text outside of box from showing up on next line.

->Text not in text box showed up next to textbox instead of below it.

Progress Report 11/14/2016

Difficult day mostly dedicated to cleaning up existing hardware/software to improve performance. Gabe showed up and helped us deal with communication issues.

Josh Passmore here most of the time. Josh Pena available every day except before 8pm on Wednesday (available 1 hour before 8pm, not much). Julia Warner available from 4p onwards every day except when there are SDP meetings.

Joshua Passmore: Soldered new addition to Power Distribution Board for servo.

Added board for additional tape sensor for ball release/target detection.

Altered design to utilize 4 bumpers instead of 3.

Finalized position of funnel.

Worked on ball release geneva wheel design.

-> Need to solidworks a way to secure servo.

->Need to do mechanical shielding for beacon.

Faux Pas: Plugged power/ground in backwards to power distribution board

-> Thankfully reverse polarity protection saved us since I had incorporated it into circuits.

Joshua Pena: Updated various states of state machine.

Implemented new beacon detection algorithm.

Implemented new beginning search.

Added new bumper for new loading ball algorithm.

Added tape sensor for releasing balls.

C problem: Somehow used more #define rather than actual numbers.

General improvements to code.

Julia Warner: Set up and uploaded html for website.
Website still needs all the links added.
->Contains code for necessary things to add.
Cutout prototype geneva wheel after filing down MDF version.
New prototype is out of foamcore.
Mounted on servo, still needs testing.
Made foam core that fits around servo to help in deciding where to place servo.
Fitted different parts of the ball loading mechanism including the funnel and exit pipe.
Exit pipe still needs to be modified more as it is still loose.
Pipe was filed down along with where it sits on the bot to allow balls through more easily.
Need to check height concerns and make sure pipe is behind depressed bumpers.

Progress Report 11/15/2016

Joshua Passmore: Made slight modifications to solidworks design for bumpers and servo mount.
Went to Ace and retrieved 2" pipes (45 and 90 deg elbows)
->Machined pipe for servo geneva wheel entry.
Attempted to lasercut new bumpers and servo mount.
->Ineffective and mount proved unusable in design.
Attempted plunger depression using sponge -> lacked necessary rigidity.
Developed mechanical shielding for beacon detector.
Soldered new power connection cable.
Attached servo and all pieces -> robot should be fully functional mechanically;electrically aside from tweaking.
Discussed possibility of 6th tape sensor -> will likely design tomorrow.
Helped detect error in location algorithm for in bounds/out of bounds sensing.
Took drill press training.

Joshua Pena: Started preparing for beer challenge.
Can now start in any orientation.
Started on any starting point algorithm.
Updated defines to include parenthesis.
Debugged homing on beacon.

Julia Warner: 1 1/2" pipe at exit of ball launching/dropping mechanism is too small for the larger ping-pong balls of 1.8".
Tried to see is small elbow pipe could be modified, but still a little too small.

Josh got a 2" 90deg pipe that seems to work really well. Top and bot needed to be cut to fit pipe and servo.
45deg pipe jutted out too much.
Servo mount and bumpers redesigned due to the constraints of the large elbow pipe.
Sponge has trouble pushing plunger on angles.
Tested estimated time for robot to make a full rotation (just under 7 sec).
Robot has some trouble backing up against ball loader.
Tested Josh Passmore's mechanical shielding for the beacon detector.
Tested Josh Pena's state machine and found some issues caused by back up algorithm.
Had issues cutting out new designs for bumpers and servo mount.

Progress Report 11/16/2016

Joshua Passmore:
Redesigned funnel to encompass larger area.
Laser cut new bumpers, funnel.
Machined crescent "geneva wheel" for ball dispensing on servo.
Determined new sensor location for ball rele tape sensor.
Installed newly cut components.
Fixed slipping motor screws.
Helped Joshua Pena debug state machine issues w/ goal tape sensor.

Joshua Pena:
Fixing up out of perimeter, not done but workable.
Up to first goal need general improvements, but on right track.
Apparently I'm a prodigy.
Started double tap.
Working on going around the goal.
Took notes for improvements.
Got to play on big field.
Celebration pizza.
Remember null pointing variables.

Julia Warner:
Not present.
Went to a meeting for Senior Design.
Meeting took 1 hour longer than expected.
Consulted with team whether to work for 2 hours then work on homework the next day or finish homework today and spend tomorrow on project.
Team said it was better to finish homework and have the full time the next day instead.

Progress Report 11/17/2016

Joshua Passmore: Soldered new isolated circuit for ball sensing.
IR sensor soldered new custom power connector.
Experimented w/ sparkfun sensor for ball detection -> ditched.
Hot glued remaining pieces.
-> May need to adjust geneva wheel slightly larger. Need to move beacon support.
Retrieved felt, hot glue, brush plunger presser and electrical tape.
Collaborated on brush placement.
Helped diagnose problems w/ state machine.
Fixed issue of wheels not turning by switching H-bridge to brake mode.

Joshua Pena: Created ball tape sensor event checker.
Added oh shit timers to find edge.
Updated backup for multiple checks.
Refactored states being used.
Created next sub state for second target.
Sensor to detect balls getting loaded set up.

Julia Warner: Decided to use a paint brush for hitting plunger.
Brush needs to be at a set angle to push in plunger.
Brush has a metal rim that Josh Passmore drilled two holes in.
The metal rim prevented brush from being flush against the back of the robot.
->Hammered in excess metal to allow brush to sit flush.
->Bent hinge to allow brush to sit at a lower angle.
->Found two screws with nuts that fit in pre-existing holes on bot.
->Filed holes to allow for sharper angles for the brush to sit.
->Fitted brush to back of bot.
Designed a larger rear bumper for back to accommodate jut of brush base.
Accidently used old SolidWorks model, so had to cut another one.

Progress Report 11/18/2016

Joshua Passmore: Got dinner for Josh.
Helped debug state machine.
Fixed height of beacon sensor.
Shifted tape sensors up to prevent colliding w/ goals.
Moved goal sensing tape sensor 1 position over.
Cycled battery once.
Sanded down demon horns.

Joshua Pena: Fixed up Double Goal/Enabled it.
Upgraded BackItUp to perform better.
Created Clear Area and ClearBeacon substates.
MinSpec without bumpers and tape sensor assurance.

Fixed up shot ball.
Added needed buffers and backs up at certain locations.

Julia Warner: Helped test robot for min-spec.
Geneva wheel sometimes drops two balls instead of one.
->Current wheel has issues blocking balls (too short)
->Took previous wheel prototype and filed it down to fit pipe.
We still need to fix obstacle avoidance.

Progress Report 11/19/2016

Joshua Passmore:

Joshua Pena:

Julia Warner: Made center hole on geneva wheel larger to be able to screw onto servo.
Helped test robot.
Collision detection needs to be more robust.
Some issues detecting bumper events when currently handling previous bumper event.
Still forgets to drop balls at targets.
Issues when dead bot is close to ball loader (still enough room for bot).

Progress Report 11/20/2016 - Onward

Debugging