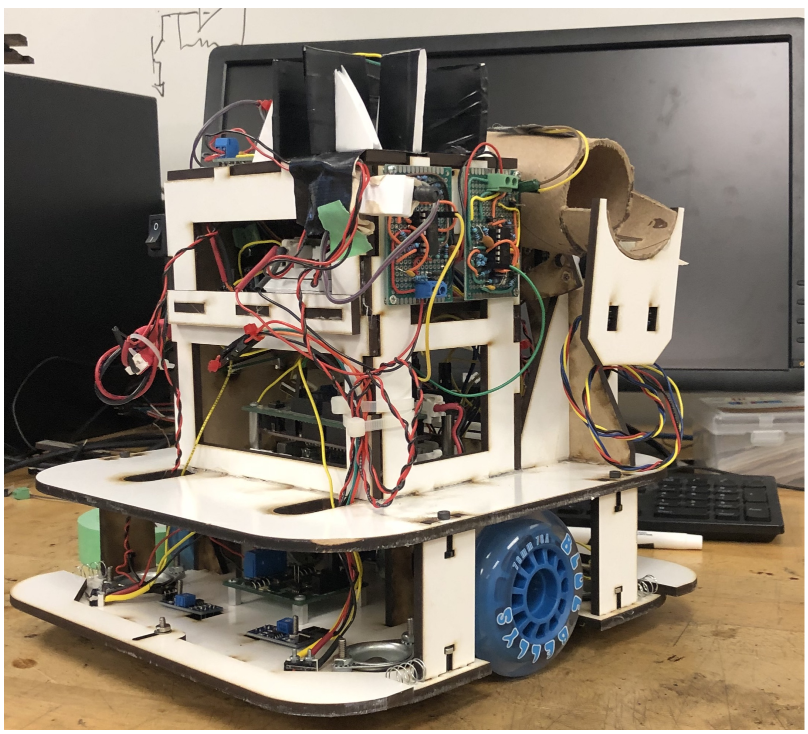
ECE118/L Final Project Report
Team "Sorry I was at an SDP Meeting"
Bot Name: GoldServo

Darius Rudominer, Katelyn Young, Nicholas Jannuzzi

June 4, 2020



Contents

Introduction

This project was designed to serve as a cumulative design project to to combine all the knowledge and skills learned in the previous labs of this class. As such, this project required the completion of an integrated system featuring mechanical, electrical, and software based systems, and utilized systems and components designed in prior labs during this quarter, such as the filter designs created in lab 1 & Perf-boarded in lab 2.

Project Background

The background of this project was to design and construct an autonomous robot capable of defeating S.A.L.T (Serial Annihilation of Living Things). Our bot was required to remotely destroy the S.A.L.T weapon by depositing ping pong balls into the three different beacon towers representing the reactor cores of the weapon. Our bot will start in a designated starting zone at the corner of the field, after which it will navigate to each of the three S.A.L.T reactors, staying within bounds and avoiding the keep out zone at the center of the field. Upon locating a reactor, the drone will locate the correct face and hole on the reactor in order to deposit the demolition charge and disable the reactor, upon which it will proceed the the next reactor.

Project Constraints and Rules

Per the course Canvas:

The field of play is a large white 8'x8' surface with a 2" black tape boundary(going out of bounds disqualifies the robot). There is one standard keep out zone on the field. It is indicated by a perimeter of 2" black tape. There are three drop off towers on the field which are marked with track wire (at the standard 24-26KHz) that runs vertically down one of the obstacle faces to the field.You will start in the "starting zone" in a random orientation.You must move to a S.A.L.T tower, locate the correct side (indicated by track wire), find the correct hole (indicated by a vertical black stripe) and deposit at least one ball in the hole. If your spy falls into the keep out zone (more than $\frac{1}{2}$ of your bot crosses the black tape) you are disqualified.If your bot leaves the field (more than $\frac{1}{2}$ of your bot outside the external boundary) you are disqualified. After depositing one ball in the correct hole on a tower,you must deposit a ball in another valid hole before earning more positive points (you can

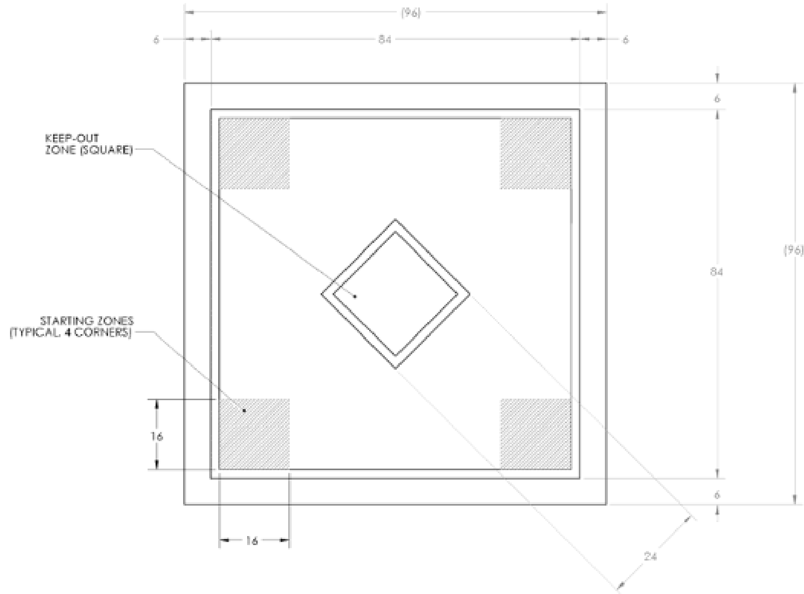


Figure 1: Field Schematic
Schematic of the playing field for the competition

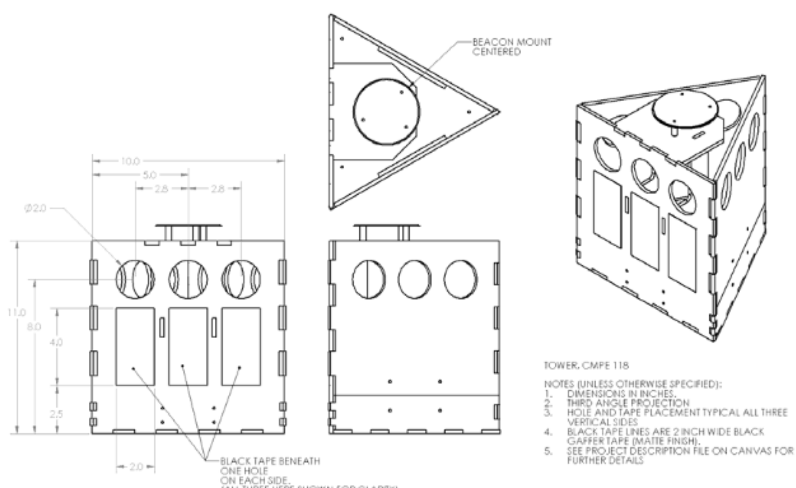


Figure 2: Tower Schematic
Schematic of a singular reactor beacon tower

deposit ten balls in a row in to the correct hole,but will only receive points for one successful drop off.)

Additional Rules:

- Bot must fit within the 11"x11"x11" cube of compliance, although the size of a bot can expand after the round begins
- Bot must be able to collision resolve within 5 seconds of contact or it will be disqualified
- Bot must remain intact throughout the match
- Sabotaging or Jamming of the opponents robot is prohibited

1 Initial Design Planning and Milestones

Our robot's layout was a result of the simplification of the design presented at the PDR during the first week of the projects. The basic premise of our bot was a trio of side mounted ball launching mechanisms, to allow our bot to simply drive up alongside the beacon and align itself with the edges of the correct face, then deposit a ball from the launcher corresponding to the correct hole. In order to do this, we planned to utilize and beacon detector capable of rotating 90°to the left, allowing us to track the beacon while performing an orbit maneuver around the tower.

Our proposed sensor suite would utilize the beacon detector unmodified from its original implementation, while two side mounted track wire detectors with analog outputs would be used to provide a method of range finding when looking for the correct wall. We chose to originally pursue using an orbit maneuver to steadily enclose upon the beacon after detecting it, using the analog output from the track wire sensors to switch to a parallel park like maneuver to get next to the wall once the correct one was found. Once close enough to the wall, dual tape sensors on either side of the magazines would help detect and align the bot with the correct hole by driving back and forth.

This design aimed to minimize the number of moving parts and by extension, the amount of control software necessary to move those parts accordingly. After discussion and better calculation of constraints, the number of magazines was revised to be a singular tube and control servo for all ball deposition, with a second servo for beacon detection, and finally tape and track wire sensing modules for navigation and hole alignment.

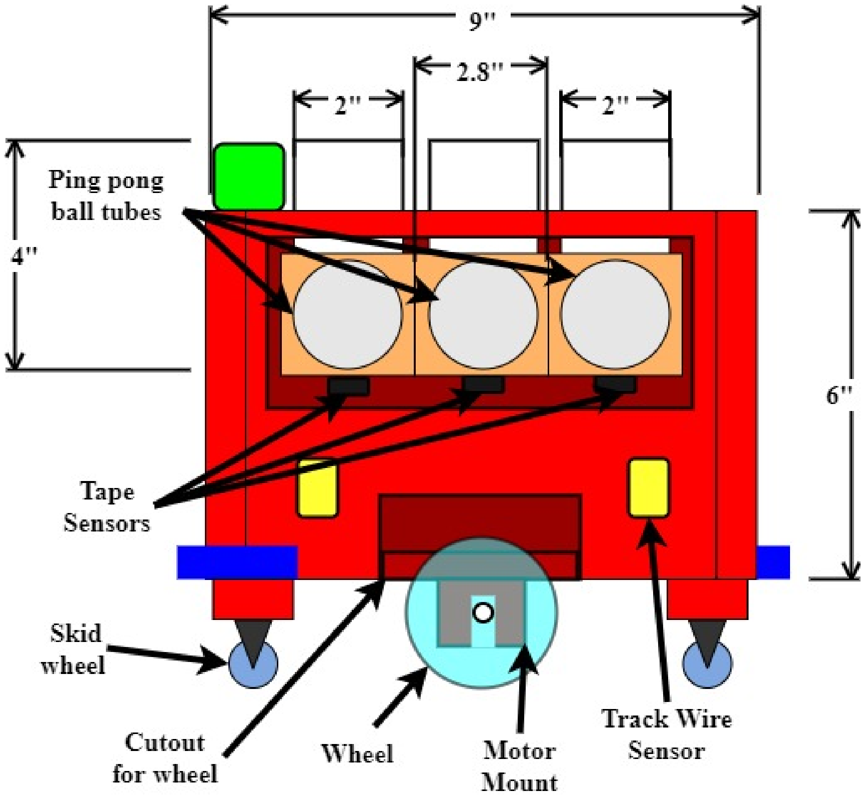


Figure 3: PDR Bot Design
Initial design sketch of the bot, featuring a triple side mounted ball dropper mechanism, later simplified to a single release mechanism

In regards to planning the design and construction of the robot, work was divided into 4 major stages, Mechanical, Electrical, Integration, & Software. As we had 4 weeks to do this project, we had a proposed schedule of accomplishing roughly one of these milestones each week. However, as we decided that time spent could be optimized by working on both mechanical and electrical components in parallel, the division of time was not exactly split into equal amounts. This allowed us to divert more time to software and any mechanical adjustments later in the project, as those were expected to be the most finicky components to deal with.

2 Mechanical Design

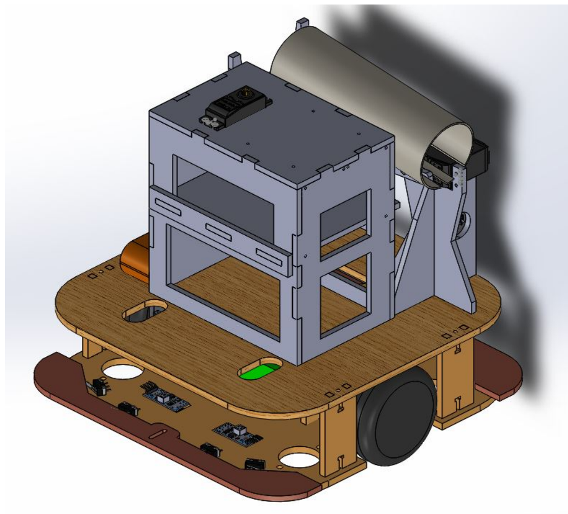


Figure 4: Isometric view of CAD model
An isometric view of the CAD model for GoldServo, missing components whose construction did not require laser cutting

With the mindset of single barreled side mounted launcher for our robot, we chose a 10x10x10” maximum size for our bot, with a base shaped like a square with rounded edges. We chose to limit our design to a smaller cube than the cube of compliance in order to give us plenty of buffer room in each dimension to extend components if necessary. A square with rounded edges provides a base that is flat on each side, allowing us to better align with the wall than a circular platform , as well as maximising usable space for mounting and reducing sharp edges that can more easily catch on sharp edges during turns. Connecting all launcher related mechanisms to a singular multi-level control tower allowed concentration of all necessary components to limit the length of wire connections need to integrate the full system. A singular fixed magazine reduces the complexity of ball dropping and shifts the heavy lifting to software, which allowed easier fine tuning of movement and ball deposition.

2.1 Motorized Lower Platform

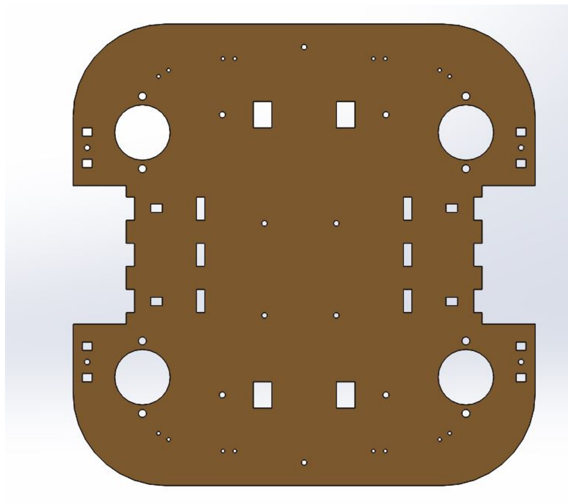


Figure 5: Bottom Plate Component
Lower plate of motorized platform with mounted components removed

We initially designed the lower platform of GoldServo to house all the necessary boards and modules for locomotion, namely the H-bridge drive and the motors themselves, along with sensors crucial for navigation around the field such as the bumpers and the bottom mounted tape sensors. Using a 10x10 square plate as the starting point, cutouts were carved to mount the wheels flush or negatively offset to allow sharp turning around the tower without worrying about the wheels catching. As a two wheeled design would rock back and forth without struts to support it, we utilized four roller wheel bearings placed evenly at each corner, using washers to ensure that they were bolted in at the right height to be level with the wheels at all times. Tab in slot with M3 screw securing mechanisms allowed us to remove the side walls and disassemble the bottom platform to access components if necessary.

2.1.1 Motor Mounting

For our motors we opted to use Greartisan 12V 60RPM DC motors purchased from Amazon. As the driving motors did not need the precision movement offered by a stepper motor, DC motors were a much cheaper and smaller choice mechanically, while requiring less software to implement properly. The specific motors we chose cam with a driveshaft offset from the center of the motor, which was specifically sought out to allow us to minimize the vertical clearance between the bottom plate and the ground and thus reduce wasted vertical space. Motor mounting wells were constructed using 2 U-shaped slots glued into tab and slot connections in the base plate, with the out mount on either side having M3 screw holes to allow the motor to be securely bolted in. Additional rib supports between the outer wall offer structural support and reduce lateral movement and stress that could damage the motor mounts and possibly separate them from the base during usage. To limit the length of the cabling from the DRV8814 driver board necessary for directional control, we opted to place the driver board directly between the motors, allowing short connection cables between them.

2.1.2 Bumpers and Switch Sensors

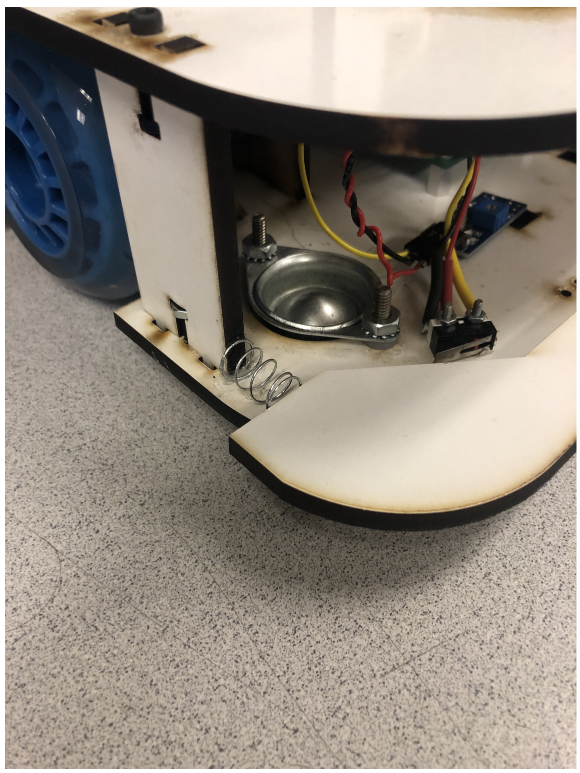


Figure 6: Bumper Push Back Springs
Springs mounted to walls to provide rebounding force on bumper component

With all obstacles present on the course having identical dimensions and at the same height, mounting of the bumpers and bumper sensors was simply a matter of putting them in a space where the movement would not be restricted by other components. As the bumper sensors we purchased were quite small with very short lever arms, there was some concern over whether they would have enough pushback to move the bumper piece after being pressed. As a precautionary measure, a second bumper was added to each of the 4 sides, in order to simply provide mechanical force to reset the bumpers, additionally, popsicle sticks were taped to the switches in order to extend the lever arm.

The bumper piece itself went through several revisions, as the initial design did not protrude far enough to reliably trigger when hitting the wall at random angles. Although this was solved by widening the bumper, we found that the bumper still failed to reliably reset even with additional switches to provide more force. Ultimately we attached springs between the support wall and the back of the bumper to provide a consistent push back force. This also had the added benefit of preventing the bumper from getting flicked when caught on a corner and triggering a false positive event on the opposite side bumpers.

2.1.3 Bottom Mounted Tape Sensors

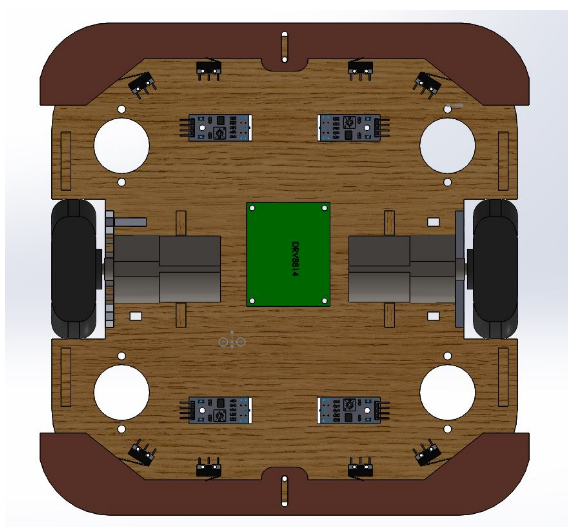


Figure 7: Bottom Plate Integrated Component
Lower plate of motorized platform with mounted components

Our lower platform utilized used 4 tape sensors, 2 on each end, in order to detect boundary tape on the field. Although the circuit needed to create the tape sensors was relatively simple, we elected to purchase a pack of 10 pre-built tape sensor boards with digital and analog output pins, as well as a potentiometer to adjust sensitivity. As these boards came with a M3 screw sized hole, they were mounted directly to the bottom plate, and slots were cut to allow the TCRT5000 reflective optical sensor to protrude through. As the tape sensors only needed to sense the presence of tape rather than the distance, we only used the digital output of the sensors.

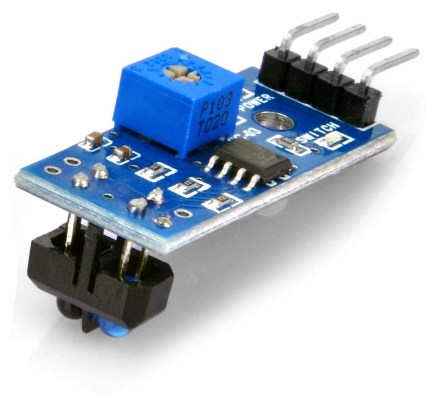


Figure 8: OSOYOO TCRT5000 module
The purchased OSOYOO tape sensor module with potentiometer

2.2 Central Control Tower

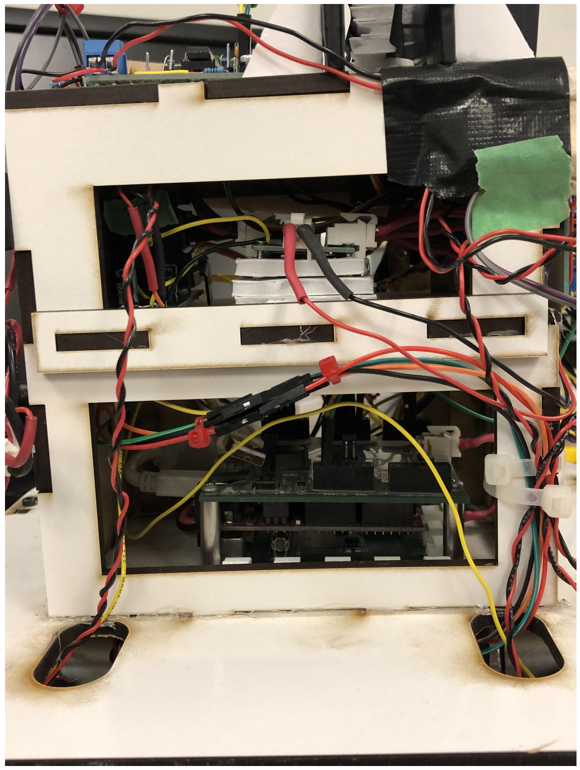


Figure 9: Front control stack view
Front window of the central tower, with Uno32 and power distribution boards visible

The central control tower served as a method of having boards with a large number of connections in one place, allowing all cabling to be routed to a single area on the bot. This was designed as a two level tower connected to the top plate, with the lower level housing the large Uno32, and the upper level holding the power regulation and distribution boards. One side of the tower held the battery and power switch, while another wall held the two track wire detector boards. The upper level was conceived as removable plate that could slide out for easy access, although we ultimately found ourselves just taking the roof off instead when changes needed to be made. Rather than construct a separate holder for the 9.9V battery, it was simply fastened to the side of the tower in a sleeve made out of a zip tie. We also added a power switch between the battery and the power boards in order to have an easily accessible switch, as the Uno32 switch was partially obstructed by the wall of the tower and proved difficult to quickly switch on an off at times.

2.2.1 Beacon Detector Aperture

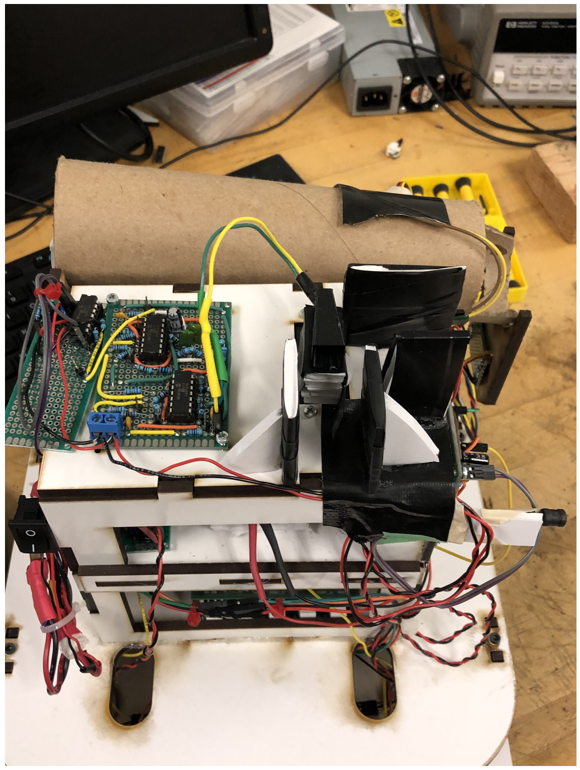


Figure 10: Top view of tower

A top view of the central control tower, showing the beacon detection board and aperture mounting

Mounted to the top of our tower was the beacon detector boards and the beacon detector aperture. As our proposed software design intended to use a rotating detector to track the tower we were orbiting as the bot moved around it, we mounted the beacon aperture to a servo. Ultimately, this method of circling the tower proved to be susceptible to detecting the other towers, so we eventually dropped the rotating design and simply used the servo as an attachment point. The aperture seen above was the result of numerous redesigns, and can be considered the 3rd iteration of the aperture. Our first design used the bottom section of a Yerba Mate can, as was mainly used to test the detection angle of the system. The second iteration used foamcore layered in black tape and plated with strips of sheet metal in order to reflect other IR sources from hitting the sensor, but this design had issues with the pinhole drooping and losing sight of the beacon. The 3rd iteration that proved to be the most effective was to utilize a custom 3D printed funnel stacked on top of layered foamcore wrapped in tape, as it was a narrow enough aperture while still be structurally secure. As the final aperture still had a wider detection angle than we liked, we added additional walls of black taped foamcore to narrow the viewing angle even further.

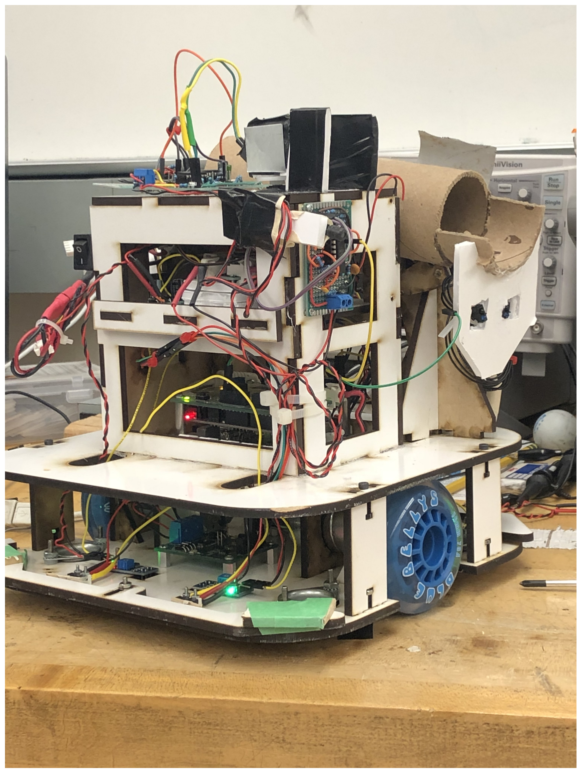


Figure 11: View of GoldServo with 2nd iteration aperture mounted
An angled view of the bot with the metal plated 2nd iteration beacon aperture mounted

2.3 Ball Deposition System

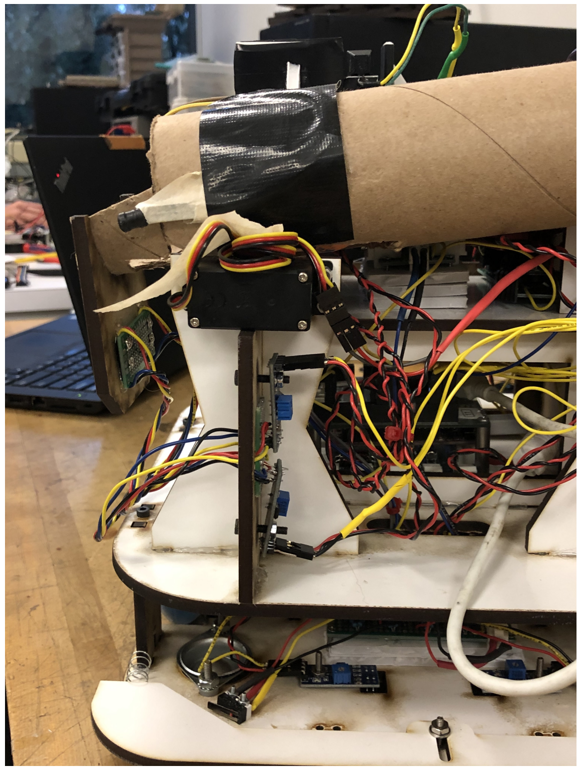


Figure 12: Side view of ball dropper
A side view of the ball deposition system, showing the tape detector mounting, servo arrangement, and tape detector extension board

Our ball deposition system consisted of a single tube of heavy duty cardboard tubing and a servo with an MDF attachment. 2 sets of interlocking pieces served as the supports struts, with the back one being taller in order to have the magazine at an angle and allow gravity feeding of ping pong balls. While we wanted to maximize the number of ping pong balls we could hold, we realized when designing the magazine that size would be an issue. Our initial design proposed using an L-shaped elbow connector to wrap the magazine around the bot carry upwards of 8-9 ping pong balls. When trying the attach the elbow piece, we struggled to find a mounting method that would ensure the bot was both structurally secure and within the size constraints. This led to us finally deciding to only carry 4 balls, as we were able to reduce the chance of missing by tuning the barrel length and the state machine instead.

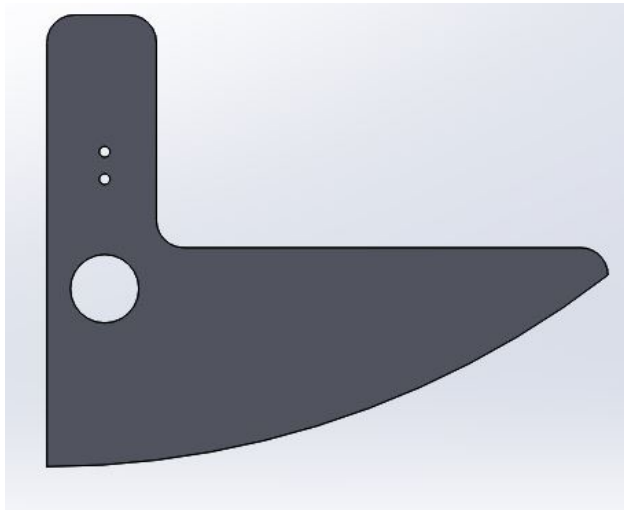


Figure 13: Side view of the servo attachment
The L-shaped servo piece used to release one ball at a time, the rounded edge pushes the others balls back when returning to its original position

The servo attachment was designed in order to eliminate the possibility of the remaining ping pong balls in the magazine from getting jammed when the servo rotates back to its starting position. The rounded, longer lower edge allows the ping pong ball that will be loaded is pushed back when rotating, after which gravity feeds the next ball into the launching mechanism.

As the magazine was not mounted flush with the side of the base of the bot, we had to extend the barrel horizontally in order to properly deposit the ball into the hole. As our bot was only 10 inches wide, we started with a half circle shaped piece of cardboard that protruded 1/2 and inch past the base, gluing it to the end of the magazine. Using cardboard rather than a laser cut design was very helpful for this, as the 1/2 inch protrusion was found to catch on the corner of towers during turn maneuvers, and so we were easily able to cut down the length of the part to resolve the catching issue without having to re-cut any parts.

2.3.1 Tape Sensor Plate

For our alignment state machine behavior we required 2 track wire sensors and 2 tape detectors. As we had design the trackwire detectors' boards to be mounted separately from the solenoids themselves via jumper wires, the boards were attached to the side of the tower, while the solenoids were mounted to a foamcore beam and the side of the magazine for front and back respectively. In order for proper detection of the correct hole, the side tape sensors needed to be mounted at a similar distance apart as the diameter of the launcher tube. Due to space concerns, we modified two tape sensors to remove the TCRT5000 sensors from the boards, attaching them to a small perfboard mechanically mounted to the MDF panel underneath the tube, and connected to the original board via jumpers. This allowed us to mount the output pins and potentiometers to the side of the support tower, allowing easier access for integration and calibrations.

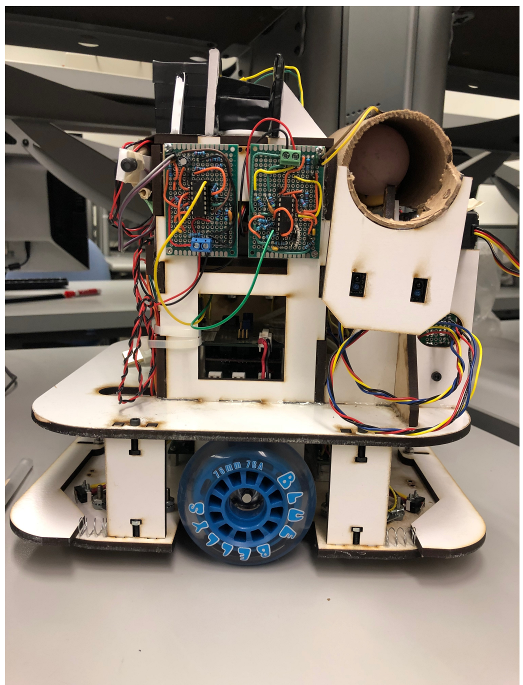


Figure 14: Side view of the ball deposition system
Track wire boards and side tape detector panel are clearly visible in this image

3 Electrical Design

The electrical design of GoldServo consisted of three major modules: motor voltage regulation, the sensor suite, and power regulation and distribution. Motor voltage regulation was implemented so that the h-bridge would be supplied with a steady voltage through operation, allowing for consistent maneuvers. The sensor suite was the bulk of the GoldServo's electrical design. This module included four different types of sensors, which will be described in greater detail in the sections below. Finally, power regulation and distribution was clearly needed to power all the different components of the system with clean consistent supplies.

3.1 Motor Voltage Regulation

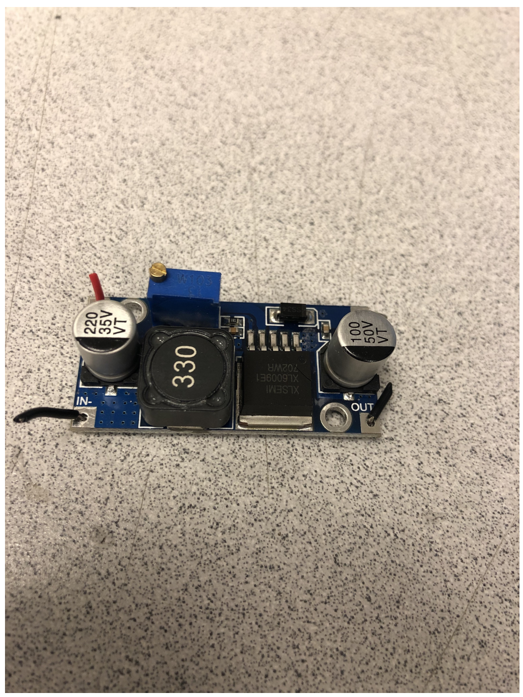


Figure 15: Buck-Boost Converter

The potentiometer adjustable Buck-Boost converter used to regulate motor voltage

An issue brought up in discussion with some of the TA's & mentors was the change in voltage output by the battery as it lost charge. As the speed of the 12V motors was affected by the amount of voltage flowing through them, this meant that our turning sped would be affected by how much charge was in the battery, meaning turn based maneuvers would not be consistent. While some mentors recommended that this could be solved in code, we chose to solve this issue using hardware, namely a small buck-boost converter purchased from another team for $3. A DC-DC converter, a buck-boost converter allows the increasing or decreasing of voltage from one magnitude to another by changing the current draw respectively. As the module purchased by our team had an adjustable range of 3-30V, this was easily set by turning a potentiometer and then gluing it in place. By always boosting up to 12V before reaching the motors, this allowed us to maintain constant voltage regardless of battery level, and had the added benefit of giving us the max speed available to the motors, as the 9.9V battery alone was well under the rated max of 12V for the motors. An initial concern when testing the functionality of this component beforehand was its wildly nonlinear behavior when less than 3.3V was input, as we observed outputs upwards of 25V. This proved to be a non issue as the 9.9V LiFE batteries were found to never be below 9V unless they were found to be damaged or over discharged.

3.2 Sensor Suite

To complete this challenge, we constructed four different types of sensors: a beacon detector, track wire sensors, bumpers sensors, and tape detectors. The beacon detector was created to monitor the field for the IR (infrared) beacons placed on each S.A.L.T tower. Two track wire sensors were created to search for the correct side of the S.A.L.T tower (the side with the active track wire). For obstacle avoidance, four bumper sensors were situated on the lower platform of the chassis. Finally, six tape detectors were implemented; four on the lower platform for tape boundary detection and two on the ping-pong ball magazine to find the correct hole. Each of these sensors, except the tape detectors which we bought, was created from scratch. We designed breadboard prototypes before we soldered the final circuits on perfboards to be mounted on GoldServo.

3.2.1 Beacon Detector

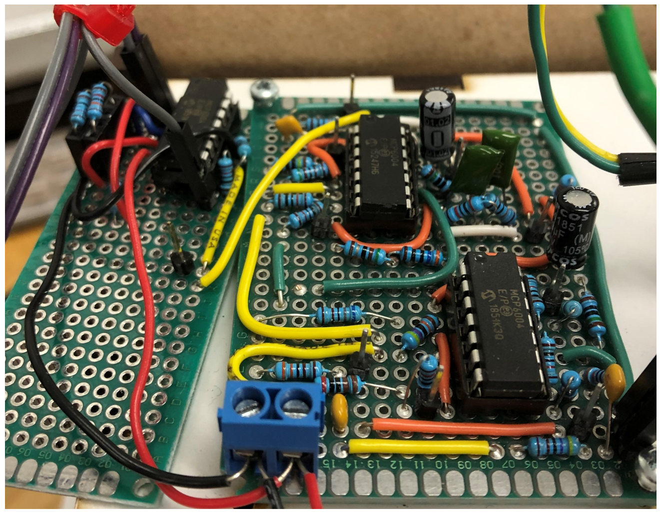


Figure 16: Beacon Detector

IR beacon detector used to search for the beacons on top of the S.A.L.T towers. Larger perfboard: original beacon detector circuit. Smaller perfboard: modifications made after connecting the sensor to the power distribution board.

The beacon detector was designed to detect 2kHz IR beacon signals through a phototransistor. We took into account the size of the field (7' by 7') and extended the range of our detection to $7\sqrt{2}' = 9.899'$ to ensure that we could find towers across the diagonal length of the field. The block diagram below displays each component of our circuit.

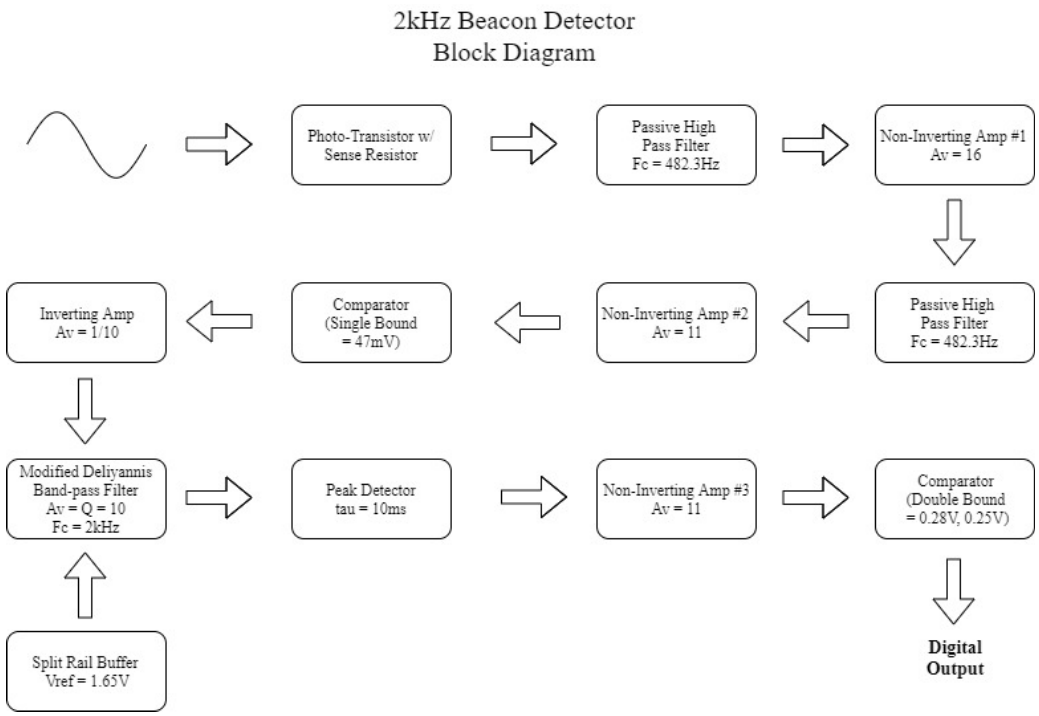


Figure 17: Beacon Detector Block Diagram
High level block diagram of the beacon detector circuit

The whole circuit can be separated into three sections: the pre-filtering stages, the modified Deliyannis bandpass filter stage, and the analog-to-digital conversion stages. The pre-filtering states were important to guarantee that our digital output would be consistent for the desired range of 9.899' and that the signal was an appropriate magnitude for the bandpass filter stage. The modified Deliyannis filter was the highlight of the beacon detector, sharply attenuating undesired IR frequencies (non-2kHz signals). The analog-to-digital conversion stage was the final piece needed to send a digital signal to the Uno32.

Pre-Filtering Stages

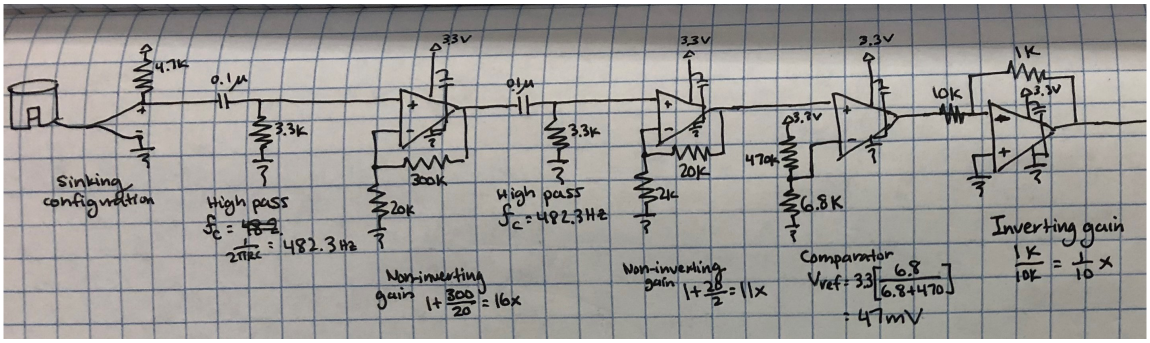


Figure 18: Pre-Filtering Stages Circuit Diagram
Detailed circuit diagram of the stages leading up to the modified Deliyannis filter

The pre-filtering stages consisted of seven parts: a photo-transistor in a sinking configuration, the first stage of passive high-pass filtering, the first stage of non-inverting gain, the second stage of passive high-pass filtering, the second stage of non-inverting gain, a single-bound comparator, and a final stage of inverting gain less than 1.

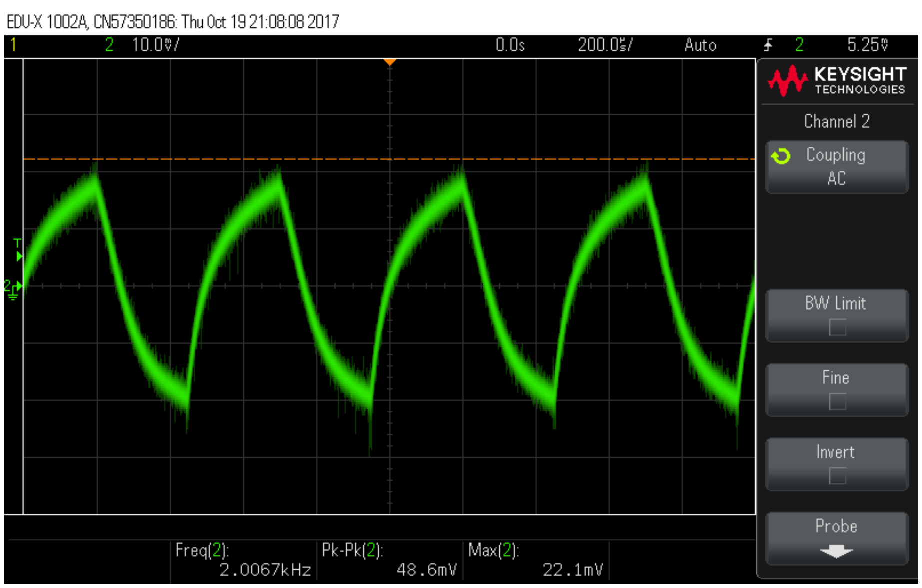


Figure 19: Photo-transistor Stage Output
Scope trace of the output of the photo-transistor stage 1.5' from the beacon

The initial signal out of the photo-transistor stage has a peak-to-peak amplitude of about 48.6 mV at 1.5' away from the beacon. To extend our detection range, we needed to amplify the signal by a large amount. However, the signal also has a noticeably high signal-to-noise ratio, due to the 60Hz noise from ambient light, as well as a significant DC bias voltage of about 3V. To avoid amplifying any unwanted noise or losing any of our signal due to biasing, we added a passive high-pass filter with a corner frequency of 483.2Hz ($R = 3.3k\Omega$, $C = 0.1\mu$; $f_c = 1/2\pi RC = 483.2Hz$). After the noise and DC bias was reduced, the signal was sent through a non-inverting amplifier of $A_v = 1 + 300k\Omega/20k\Omega = 16$.

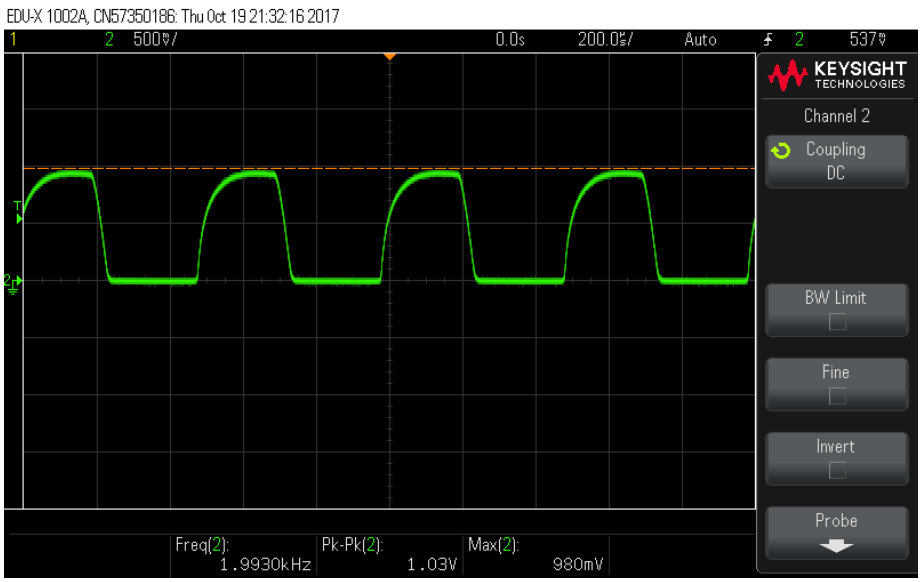


Figure 20: First Gain Stage Output

Scope trace at the DC-coupled output of the first gain stage 1.5' from the beacon

We noticed that there was another significant DC bias of about 0.5V due to the amplifier gaining the small DC bias not filtered out by the high-pass filter. To remove this bias, we added another stage of passive high-pass with the same 483.2Hz corner frequency. This resulted in a signal with a peak-to-peak amplitude of about 44mV at 9' and 0.99V at 1.5'. While this range of values might have been sufficient, we decided to extend the detection range to be longer than the required 9.899' to ensure that we would not miss any beacons. So the signal was sent through a second non-inverting amplifier of $A_v = 1 + 20k\Omega/2k\Omega = 11$.

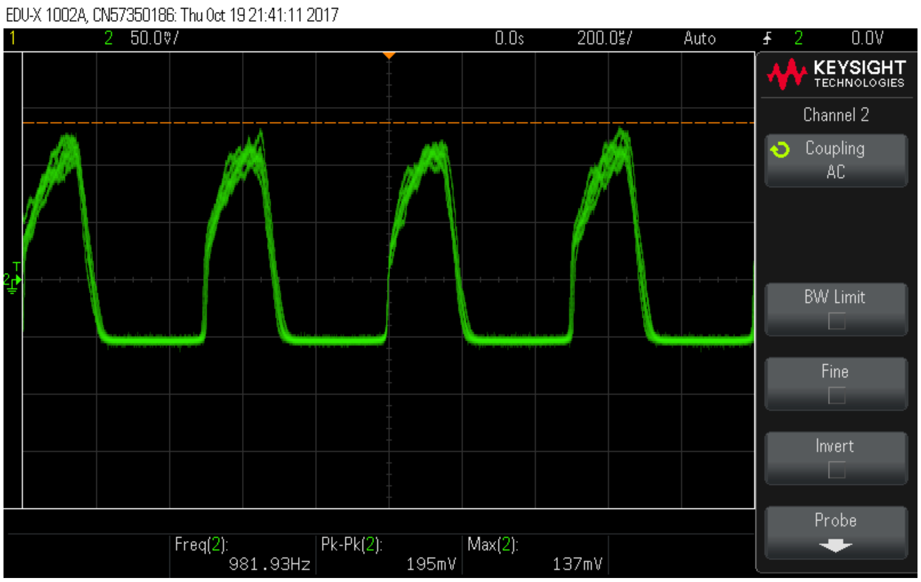


Figure 21: Second Gain Stage Output at 9'

Scope trace at the output of the second gain stage 9' from the beacon

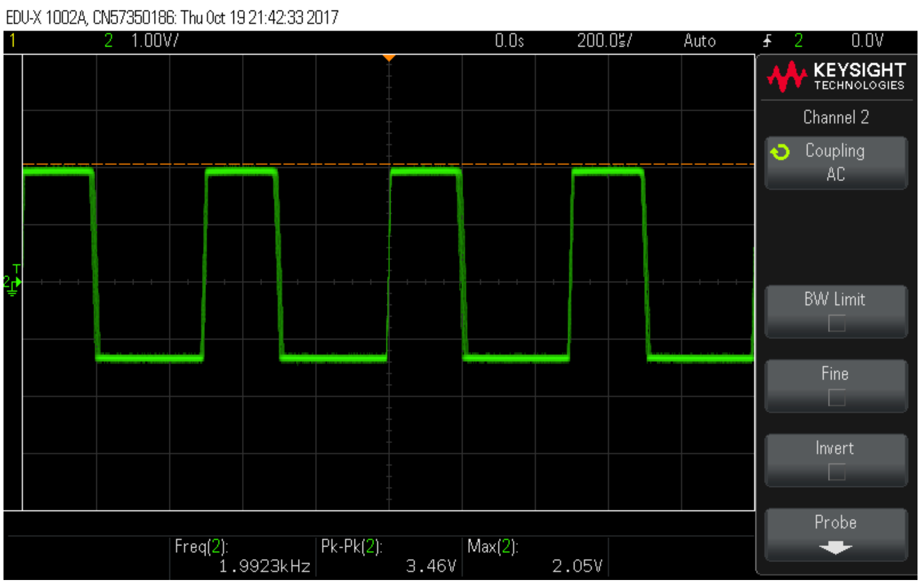


Figure 22: Second Gain Stage Output at 1.5'

Scope trace at the output of the second gain stage 1.5' from the beacon

After the second gain stage, we added a comparator with a single bound at $V_{ref} = 3.3V[6.8k\Omega/(6.8k\Omega + 470k\Omega)] = 47mV$. Thus any 2kHz signal with a magnitude larger than 47mV after the gain stages will result in a railed (3.3V) square wave. This means that we will have the same signal regardless of our distance from the beacon. By removing the distance factor from the circuit, less overall filtering is needed to distinguish between 2kHz and undesired signals. Finally, we needed to reduce our 3.3V square wave signal significantly before sending it into the filter stage. Since the modified Deliyannis filter has inherently high gain (Gain = Q = 10), we needed to reduce the incoming signal to avoid railing the signal on the output of the filter and making it impossible to distinguish between 2kHz and undesired signals. Thus, we send the comparator output into an inverting amplifier with $A_v = -1k\Omega/10k\Omega = -0.1$. This final de-amplified signal was then sent through the modified Deliyannis filter.

Modified Deliyannis Filter

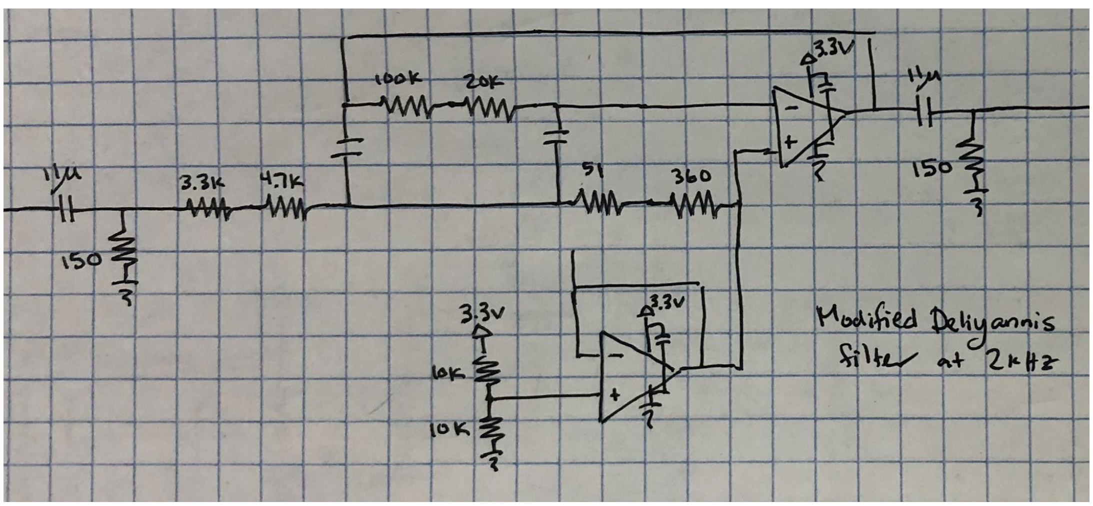


Figure 23: Modified Deliyannis Filter Circuit Diagram
Detailed circuit diagram of the modified Deliyannis filter

The filter we chose was a Texas Instruments second order single frequency band-pass filter called a modified Deliyannis filter. We chose this filter as it provided us with sharp attenuation on non-2kHz signals, while only requiring relatively few circuit components and two op amps. We designed the filter based on the following equations found in two Texas Instruments design documents: *Filter Design in Thirty Seconds* and *More Filter Design on a Budget.*

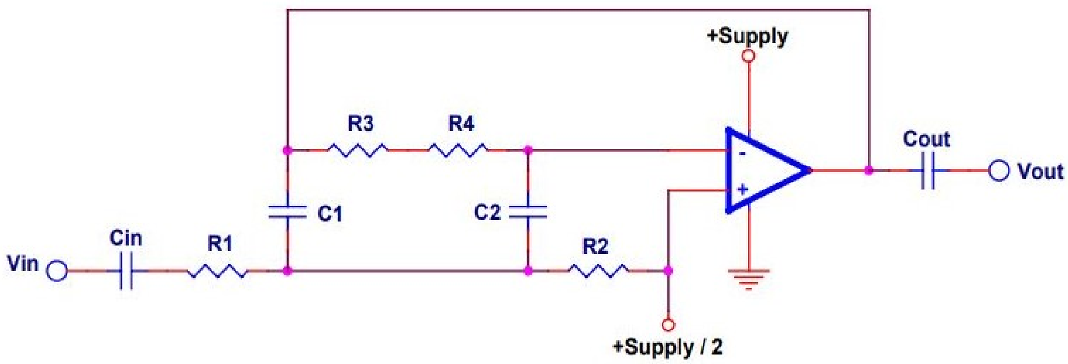


Figure 24: Modified Deliyannis Filter Circuit Template
Template of circuit diagram of the modified deliyannis filter from Filter Design in Thirty Seconds

$$C_1 = C_2 = 10nF$$
$$R_1 = R_4 = \frac{1}{2\pi c_1 f} = \frac{1}{2\pi(10nF)(2kHz)} = 7.96k\Omega$$
$$R_2 = R_1/19 = 7.96k\Omega/19 = 418.95 \;\text{–>}\; 51\Omega + 360\Omega = 411\Omega(actual)$$
$$R_3 = 19R_1 = 19(7.96k\Omega) = 151.24k\Omega$$

Although the circuit was designed for a 2kHz center frequency, we were not sure that the actual circuit would produce the desired results. Thus, we decided to use two potentiometers (10kΩ and 200kΩ) for R_1 and $R_3 + R_4$ to fine-tune the filter. We inputted a 200mVPP AC signal from the signal generator directly to the filter to test the circuit before connecting it to the previous pre-filtering stages. The table in figure 25 shows the peak-to-peak voltages for frequencies 1.5kHz-2.5kHz for one and two stages of the filter.

frequency	one stage [VPP]	two stages [VPP]
1.5	0.084	0.19
1.6	0.109	0.26
1.7	0.141	0.44
1.8	0.193	0.84
1.9	0.293	1.93
2	0.4	3.3
2.1	0.3	1.93
2.2	0.22	0.92
2.3	0.165	0.53
2.4	0.133	0.36
2.5	0.117	0.26

Figure 25: Peak-to-peak voltage table of filter outputs
Peak-to-peak output voltages of modified Deliyannis filter for one and stages at frequencies from 1.5kHz-2.5kHz

At 1.5kHz, there was a 13.56dB drop for one stage and 25.8dB drop for two stages. At 2.5kHz, there was a 10.7dB drop for one stage and 22.1dB drop for two stages. Using two stages of filtering are clearly better at attenuating undesired frequencies, but we did not need that significant of a dB drop to capture the difference between 2 kHz and 1.5/2.5 kHz with the peak detector and comparator. This significantly reduced the amount of circuit components required and saved us time when we had to solder the final circuit.

Further modifications were made to this filter stage after the analog-to-digital conversion stages were constructed to test if we replace two potentiometers with regular 1% resistors. Because there are no beacons with "bad" frequencies (not 2kHz) close to 2kHz, our filter does not need to have a peak at exactly 2kHz. Thus, we were able to swap out the 10kHz potentiometer for a 3.3kΩ resistor in series with a 4.7kΩ resistor and the 200kHz potentiometer for a 100kΩ resistor in series with a 20kΩ resistor. This modification significantly cut down on the space required when we soldered the final circuit. The resulting filtered signal at 1.5' away from the beacon is displayed in figure 22.

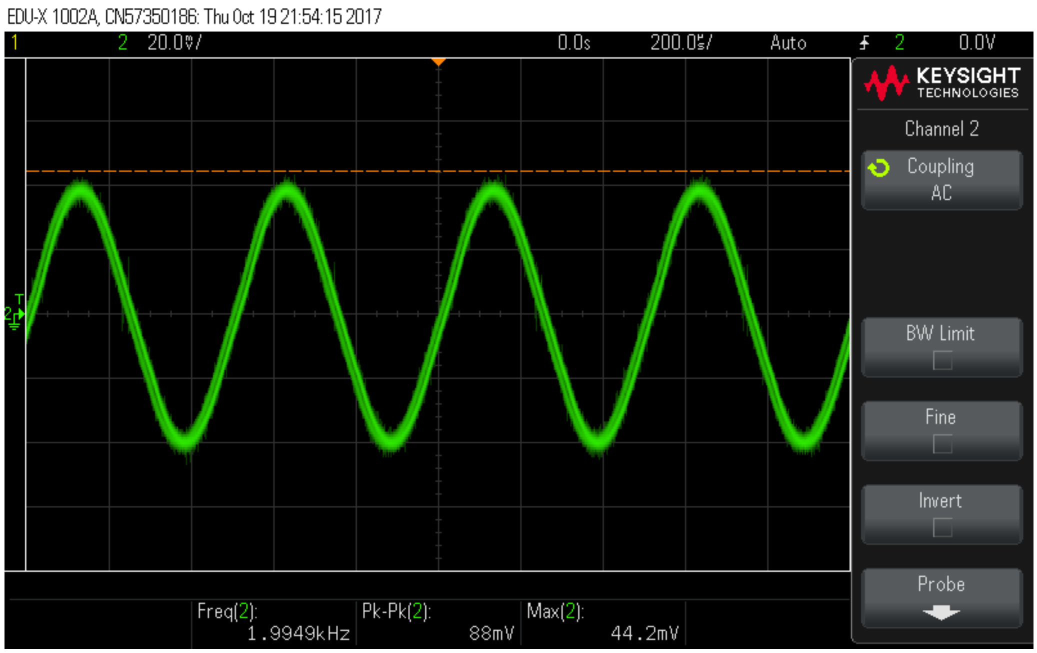


Figure 26: Filter Stage Output at 1.5'

Scope trace at the output of the modified Deliyannis filter 1.5' from the beacon

Analog-to-Digital Conversion Stages

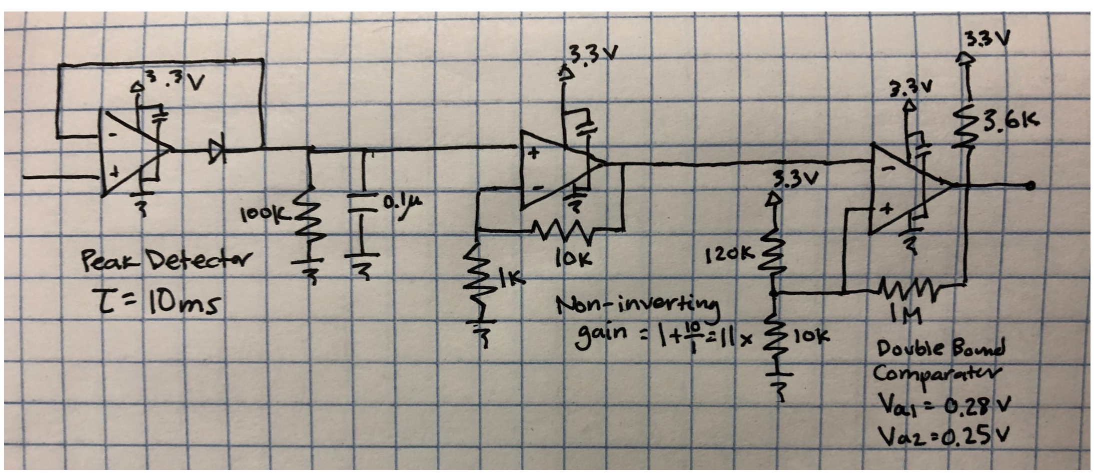


Figure 27: Analog-to-Digital Circuit Diagram

Detailed circuit diagram of the stages following the modified Deliyannis filter

The analog-to-digital conversion stages consisted of three parts: a peak detector, a single stage of non-inverting gain, and a comparator with hysteresis. The peak detector captures the maximum voltages of the filtered signals through the capacitor, creating semi-digital signals. The resistor allows the charge to dissipate to allow the circuit to distinguish between voltage peaks. We chose the 100kΩ resistor and $0.1\mu F$ capacitor for a time constant of 10ms to work with the 2kHz signal.

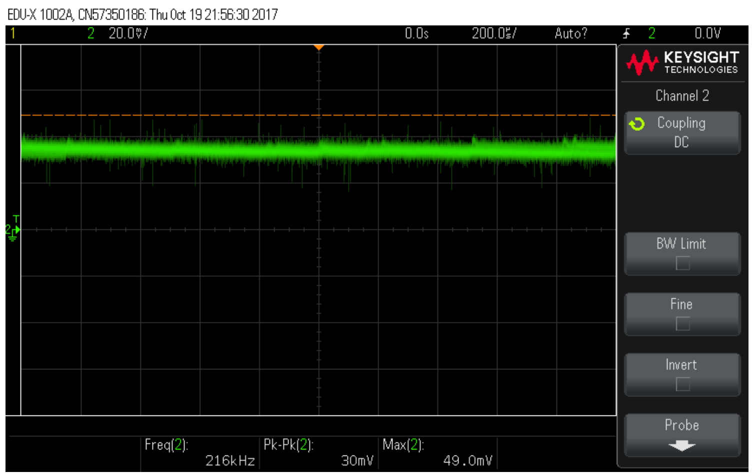


Figure 28: Peak Detector Output with the beacon in sight
Scope trace at the output of the peak detector with the beacon in sight

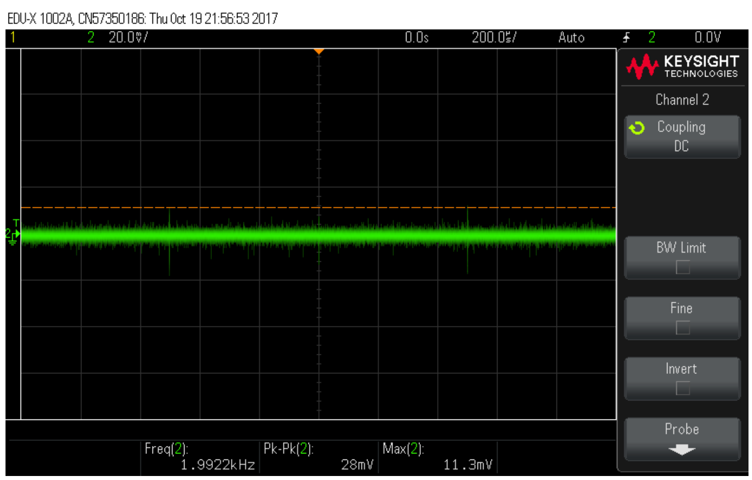


Figure 29: Peak Detector Output with the beacon out of sight
Scope trace at the output of the peak detector with the beacon out of sight

With the current version of the power distribution board, the peak detector output is at about 50mV when the beacon is in sight, and about 11mV otherwise. The original version of this circuit consisted of only a comparator with a single bound at $V_{ref} = 3.3V[5.1k\Omega/(5.1k\Omega+1M\Omega)] = 16.7mV$. This bound worked when we isolated the beacon detector from GoldServo, connecting it to the lab power supply, instead of GoldServo's power distribution board. However, our original version of the power distribution board introduced a large amount of noise on the 3.3V power rail. Since all our other sensors were functioning even with this noise, we originally decided to change our comparator bounds instead to compensate for the noise. For a more robust design, we switched to a comparator with hysteresis (inverted configuration). The component values were determined by the following equations.

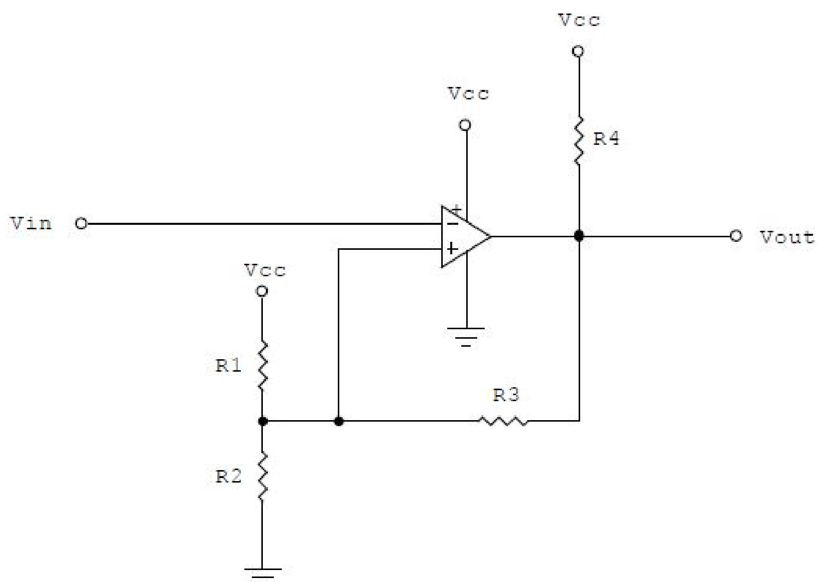


Figure 30: Comparator with Hysteresis Circuit Template
Template of the circuit diagram of a comparator with hysteresis from the ECE 118 Lab 1 manual

$V_{a2} = \textit{lower trip point} \qquad V_{a1} = \textit{higher trip point}$

$R_4 = 3.9k\Omega \;->\; 3.6k\Omega \;(\textit{standard}) \qquad R_3 = 1M\Omega$

$$R_1 = \left(\frac{V_{a1} - V_{a2}}{V_{a2}}\right) R_3 \qquad R_2 = \left(\frac{1}{R_1} + \frac{1}{R_3}\right)^{-1} \left(\frac{V_{cc}}{V_{a1}} - 1\right)$$

We had five different versions of comparator values as the maximum and minimum output values of the peak detector changed with different light environments and different mechanical shielding. The bound were determined by using the multimeter to measure the peak detector DC voltage output when the beacon was in sight and out of sight. We then set the upper and lower bounds such that the in-sight voltage was higher than the upper bound and the out-of-sight voltage was lower than the lower bound. The table shows all the versions of the comparator values with their various conditions.

Version	R1 [kΩ]	R2 [kΩ]	Va1 [V]	Va2 [V]	Conditions
1	330	3	0.0394	0.0296	in daylight
2	120	6.2	0.1805	0.1612	on breadboard w/gain
3	82	6.8	0.2717	0.2511	on perfboard at night
4	330	24	0.2910	0.2188	in fab lab (2nd shielding)
5	120	10	0.2817	0.2515	current version

Figure 31: Comparator values table
This table displays the different versions of the comparator w/ hysteresis with the current version highlighted in yellow

This modification was made after the original circuit was already soldered, so we used a pin at the peak detector output to connect the previous stages to a breadboard prototype of the comparator circuit. We found that the first version worked during daylight hours. After verifying the expected digital output — $\sim 0V$ when beacon detected, 3.3V (railed) when not detected — we soldered the comparator circuit onto another smaller perfboard (see Figure 16). To ensure that we could change R_1 and R_2 if the comparator required tuning, we used a 1 Row 4 Position Vertical Female Pin Header for each resistor.

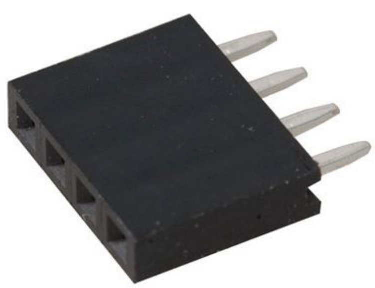


Figure 32: Row of Female Pin Headers
Jameco part# 2169758: 1 Row 4 Position Vertical Female Pin Header

After soldering the circuit, we checked the scope output again for verification. We found that the digital output was not entirely reliable, as the bounds were extremely close to the noise threshold. We found it difficult to find more appropriate bounds with the given standard 1% resistors. Thus, we decided to add a non-inverting gain stage with $A_v = 1 + 10k\Omega/1k\Omega = 11$ to increase the difference between the maximum and minimum values of the peak detector. This would give a wider range of values for possible comparator bounds. This was very useful in dealing with standard resistor values, which required us to have a larger margin of error between the calculated resistor values from the given equations and the actual values implemented.

Version two included the bounds for this new setup — breadboard prototype and final perfboard design. This version worked until lighting conditions changed. Version three shows the bounds at night. This circuit worked until the next day when we moved to the "fab lab" (the lab with the S.A.L.T field). The lighting conditions in here were different than in the lab where we were constructing the bot. However, no windows are present in the "fab lab," so this was the final lighting environment that our robot needed to adjust to. We should have calibrated our sensor for this lab initially, but there was no oscilloscope in this room, which made it difficult to perform any extensive debugging.

Version four includes the bounds for this new setup, which used the second iteration of the mechanical shielding mentioned above in section 2.2.1. This version worked for a couple days, but eventually the noise threshold increased to a point where we decided to mitigate the noise instead. This process will be described in detail in section 3.3. After eliminating the noise, we found new bounds for the final setup, shown in version five. The figures below show the scope traces from the output comparator when the beacon is in and out of sight.

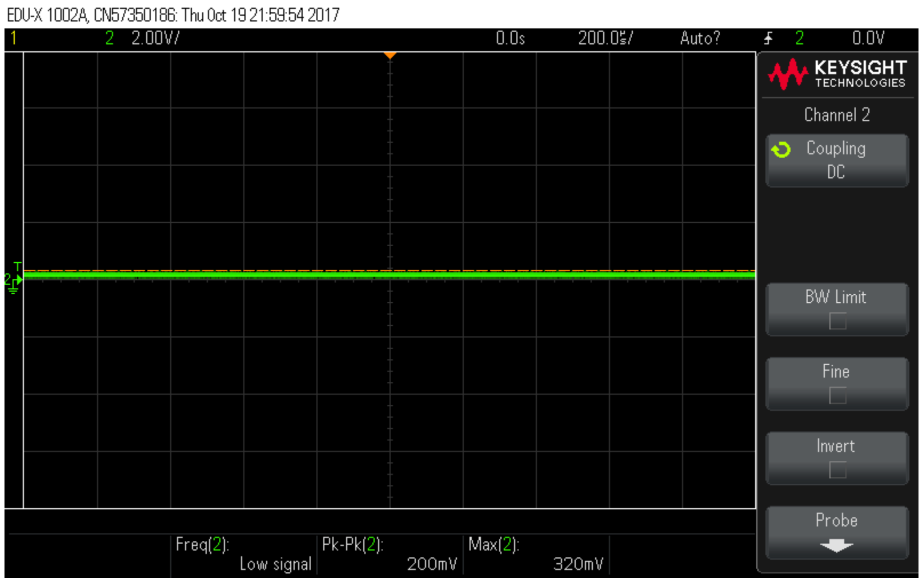


Figure 33: Comparator Output with the beacon in sight
Scope trace at the output of the comparator with the beacon in sight

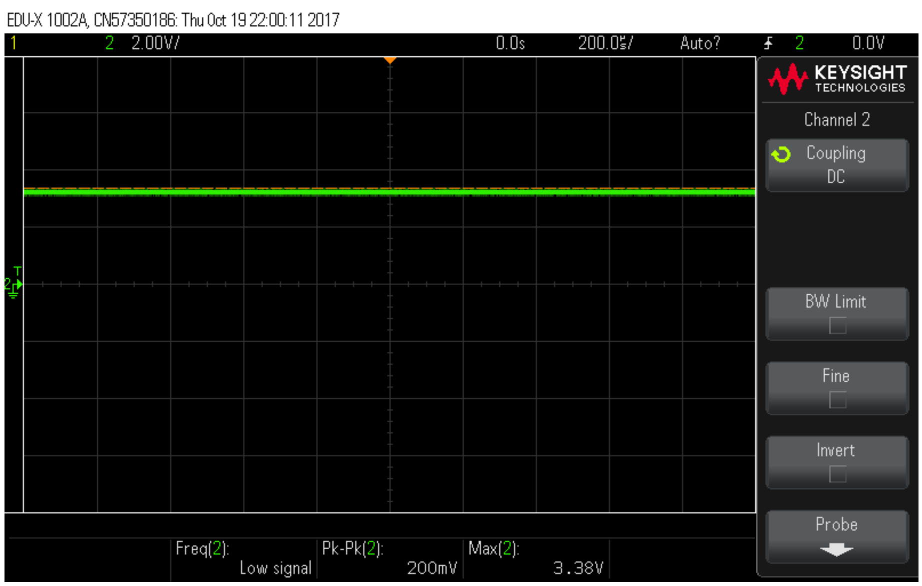


Figure 34: Comparator Output with the beacon out of sight
Scope trace at the output of the comparator with the beacon out of sight

Soldering the Beacon Detector

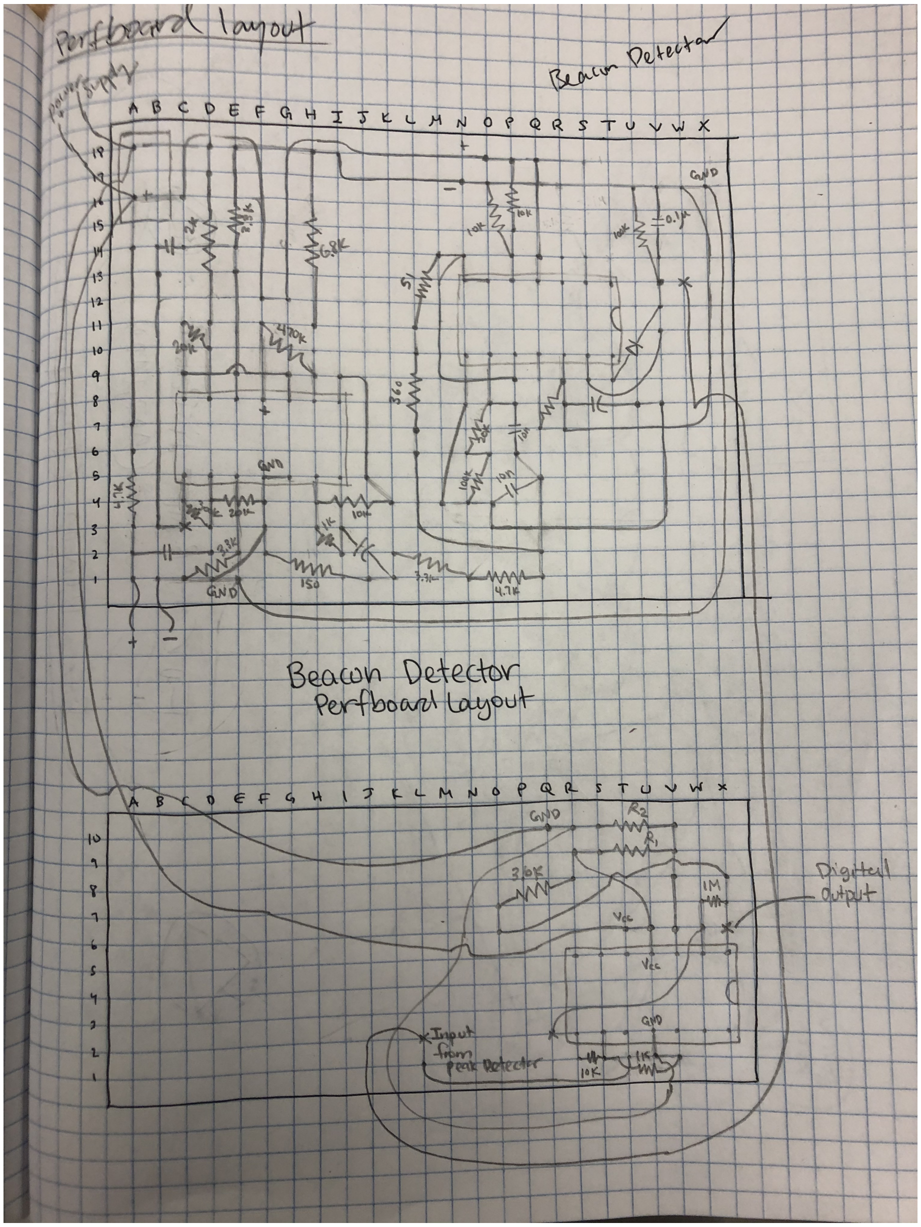


Figure 35: Perfboard Layout of the Beacon Detector

Perfboard layout of the beacon detector with the original circuit on the top perfboard and the modifications on the bottom perfboard

The perfboard layout of the final beacon detector design is shown on the previous page with the original circuit on the top perfboard and the modifications on the bottom perfboard. We first started with soldering a blue screw terminal for power and ground and the required MCP6004 ICs. Our original circuit only needed eight op amps, so we soldered in two ICs. Our layout plan was designed such that we would use the least amount of wires, would avoid overlapping components, and would have easily accessible power rails. Creating the layout was quite difficult and time-consuming, but it made the soldering process proceed smoothly.

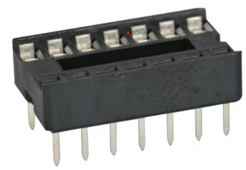


Figure 36: 14-pin IC Socket
Jameco part# 112214: Socket IC 14-Pin Dual Wipe Low Profile 0.3 Inch Wide

14-pin IC sockets were implemented for mounting/dismounting ICs without damage to the perfboard or the pins themselves. We employed the method of using long component leads for short connections and wires for longer connections. Each module of the circuit was built one by one, in the order shown in the block diagram at the beginning of this section, starting with the photo-transistor circuit. Each section was created by: placing the components in the correct holes according to the layout, taping the components flush to the perfboard, soldering the leads/wires to the solder pads, bending the leads/wires to form connections, and creating the solder joints. Male header pins were soldered at the outputs of each stage for testing purposes. Continuity tests were conducted with the creation of each solder joint. After verifying the connection, we clipped off any overhanging component leads. We verified the expected output on the oscilloscope of each section with a test beacon before beginning the next section. Since we created a solid initial layout, were careful in forming each connection, and tested each stage of the circuit, no mistakes were committed and no de-soldering was required.

For the modifications to the circuit, we needed two more op amps, so we decided to solder another IC socket and the female header pins described earlier onto another smaller perfboard. The new circuit was connected to the original circuit using soldered-in wires. We could have used male header pins, but we wanted to ensure that there were no loose connections.

3.2.2 Track Wire Sensor

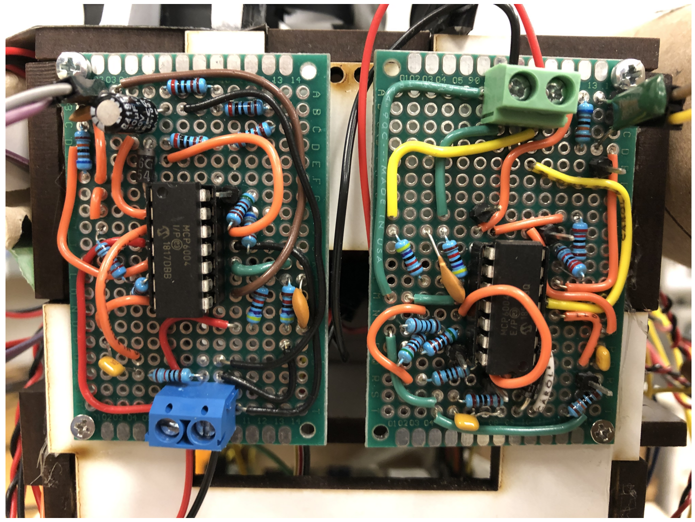


Figure 37: Track Wire Sensors
Track wire sensors to detect the correct face of the S.A.L.T tower. The left perfboard is the front track wire sensor, while the right perfboard is the back track wire sensor.

The track wire sensors were designed to detect the active face of a S.A.L.T tower via a track wire on that carries a current oscillating at 24-26kHz at distances of approximately 2”. The block diagram below displays each component of our circuit.

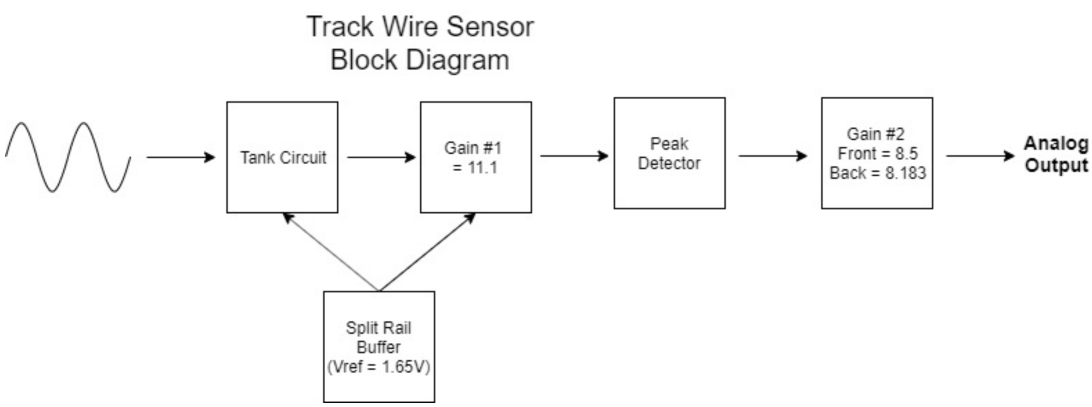


Figure 38: Track Wire Sensor Block Diagram
High level block diagram of the track wire sensor circuit

This system consists of five different sections: a tank circuit, a split rail buffer, a stage of non-inverting gain, a peak detector, and a second stage of non-inverting gain. A circuit diagram of the system is displayed below.

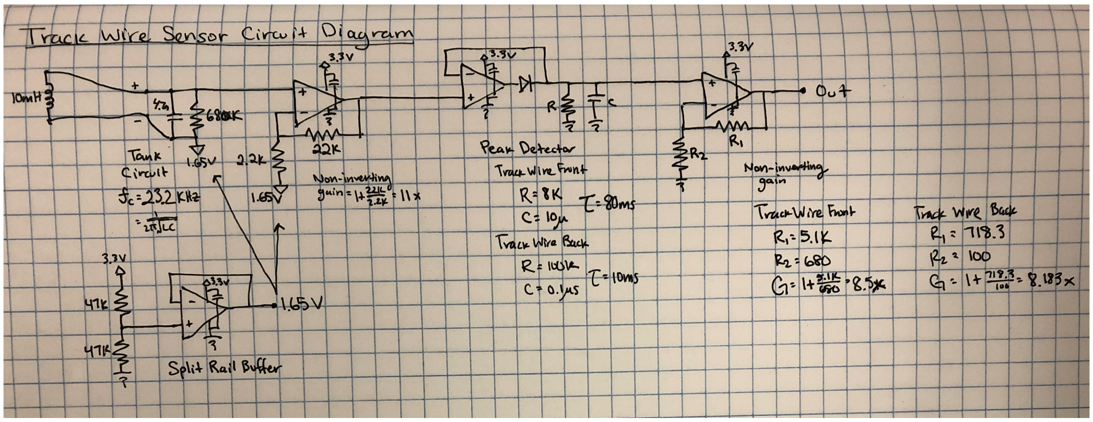
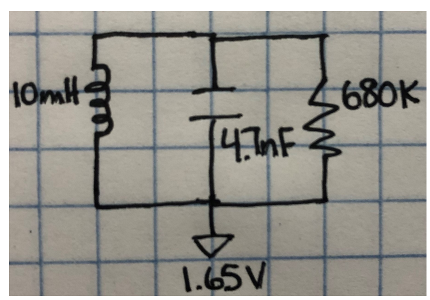


Figure 39: Track Wire Sensor Circuit Diagram
Full circuit diagram of the track wire sensor

Tank Circuit



The first section of the system is the tank circuit, which is an RLC oscillator. The current running through the track wire creates a magnetic field that induces an oscillating EMF voltage in the inductor/solenoid of the tank circuit. When the resonant frequency of the oscillator is set close to the frequency of the track wire current, the output voltage signal is amplified. We chose a 10mH inductor and 4.7nF, which corresponds to a corner frequency of $f_c = 1/2\pi\sqrt{LC} = 23.2kHz$. Since our target frequency is about 25kHz, the 680kΩ resistor is incorporated to increase the bandwidth. At a distance of 1" inches from the track wire, the resulting waveform has a peak-to-peak amplitude of about 28.5mV. At 2" inch away, the peak-to-peak amplitude is about 23.3mV. The circuit is referenced to 1.65V, instead of ground. This ensures that the entire signal, positive and negative voltages, will not be lost. The source of this voltage is the split rail buffer, which is described next. The resulting scope traces at 1" and 2" are shown below.

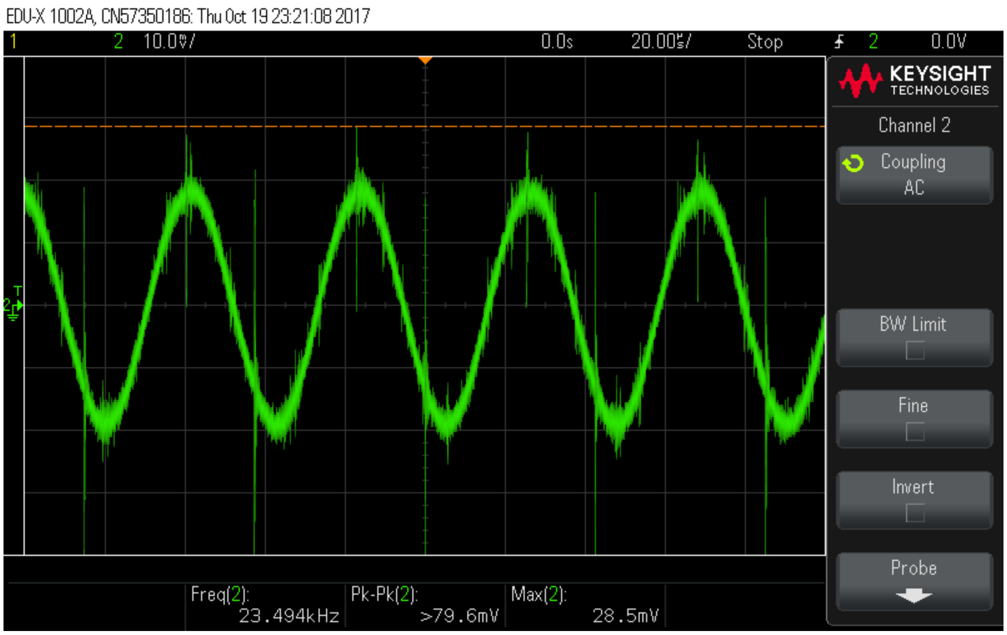


Figure 40: Tank Circuit Output 1" away

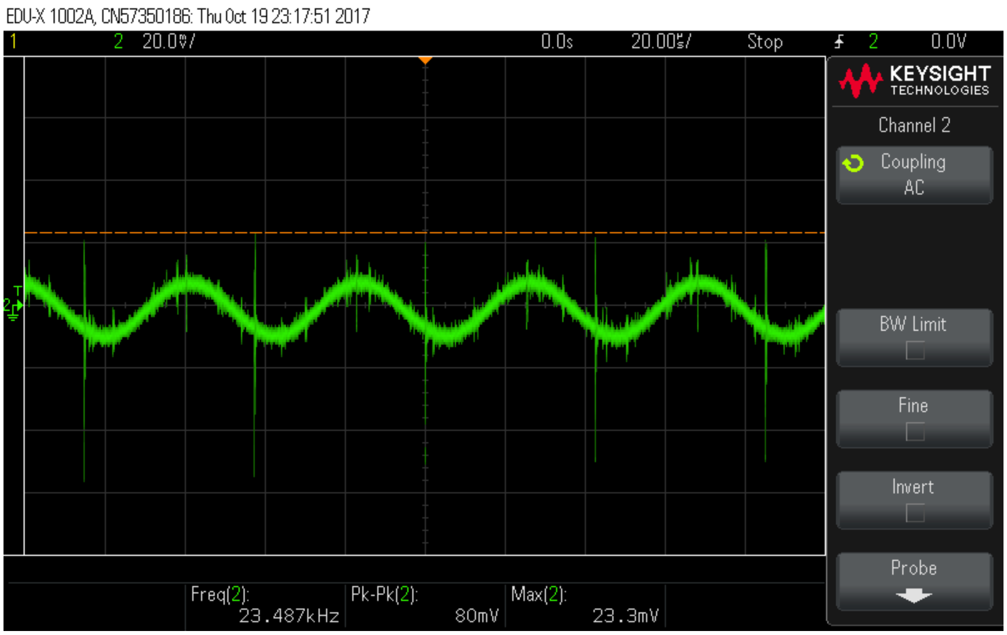
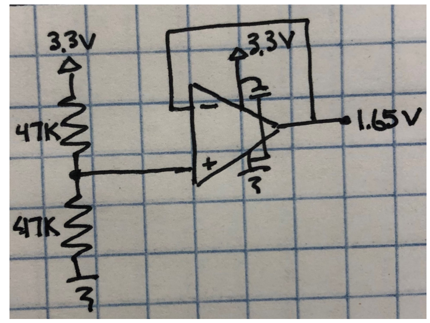


Figure 41: Tank Circuit Output 2" away

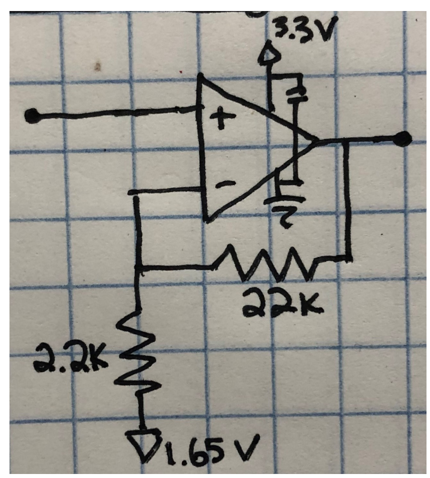
Split Rail Buffer

The split rail buffer generates a DC offset reference voltage during the non-inverting amp stage. The op amps are all single supply, so the split rail buffer allows for the positive and negative sides of the oscillating voltage. While a simple voltage divider could have been used to provide a reference voltage, this would make the system susceptible to current draw. Adding the buffer after the voltage divider eliminates this possibility. Since the target voltage was 1.65V and the source voltage was 3.3V, the two resistors needed to be equal or very similar. Thus, we chose two 47kΩ resistors. The output of this circuit is be the reference for both the tank circuit and the first stage of non-inverting gain.



First Non-Inverting Gain Stage

The induced voltage in the tank circuit has a very low magnitude, making further signal manipulation difficult. The non-inverting amplifier simply gains the tank circuit signal to a reasonable voltage. A gain of $A_v = 1 + 22k\Omega/2.2k\Omega = 11$ was sufficient. At 1" away from the track wire, the resulting waveform has a peak-to-peak amplitude of about 186mV. At 2" inch away, the peak-to-peak amplitude is about 111mV. At 4" inch away, the peak-to-peak amplitude is about 27mV. The circuit is referenced to 1.65V, instead of ground. Since the tank circuit was referenced to 1.65V, this gain stage needed to have the same reference to function properly. The resulting scope traces is shown below.



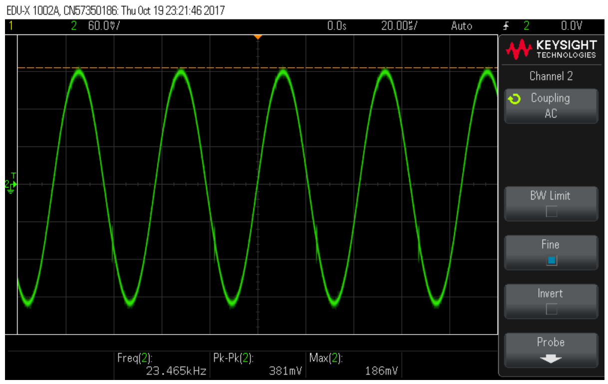


Figure 42: First Gain Stage Output 1” away

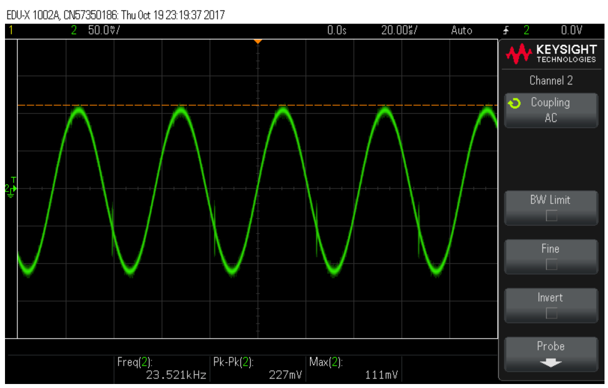


Figure 43: First Gain Stage Output 2” away

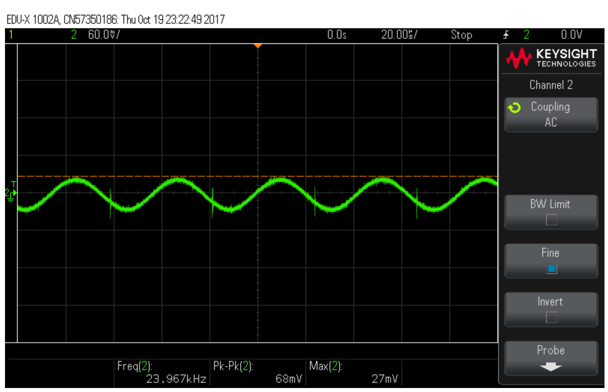
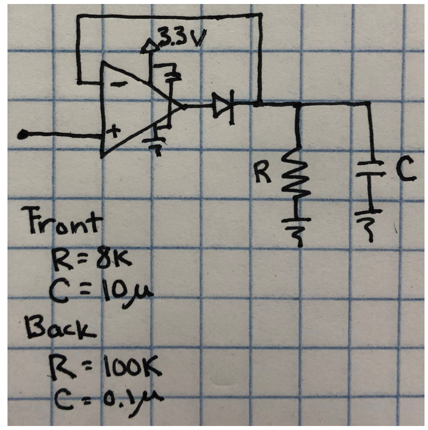


Figure 44: First Gain Stage Output 4” away

Peak Detector

At this point, we had an adequate AC signal indicative of the track wire being within 2 inches. Unlike the beacon detector, we wanted our track wire sensors to output an analog signal so that we could map a value to the distance from the track wire. This enabled us to tell us the difference between about 2" away from the tower (for finding the active face) versus less than 1" away from the tower (for parallel parking). We used a peak detector to generate this analog signal. The diode in the circuit acts as a switch that only allows the positive voltages to flow through and rejects negative voltages, rectifying the signal. The capacitor charges up to the maximum voltage, capturing the "peak." Adding the resistor allows the capacitor to discharge and catch future peak voltages. Thus, we generated a mostly-DC output at a level equal to the amplitude peak of the AC signal. The DC output increases as the sensor moves close to the track wire and decreases as it moves away.



The values for the two components are determined by the desired RC time constant. We wanted this value to be fairly small to increase the response time. Our front and back track wire sensor circuits consisted of different components, which resulted in different $\tau = RC$ time constants.

<u>Front Track Wire</u>

$R = 8k\Omega$ $\quad$ $C = 10\mu$ $\quad$ $\tau = RC = 80ms$

<u>Back Track Wire</u>

$R = 100k\Omega$ $\quad$ $C = 0.1\mu$ $\quad$ $\tau = RC = 10ms$

This was due to the sensors being constructed by two different team members. However, both sensors produced similar signals. At 1" away from the track wire, the resulting waveform for the back sensor has an amplitude of about 159mV. At 2" inch away, the amplitude is about 96mV. At 4" inch away, the amplitude is about 58mV. The resulting scope traces is shown below.

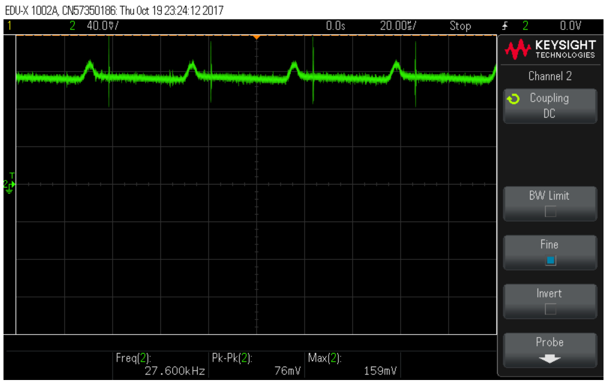


Figure 45: Peak Detector Output 1” away (back sensor)

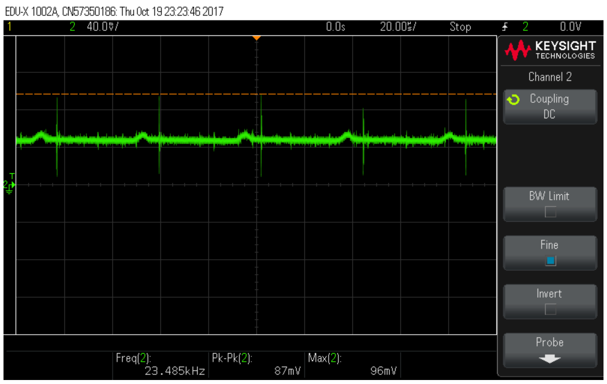


Figure 46: Peak Detector Output 2” away (back sensor)

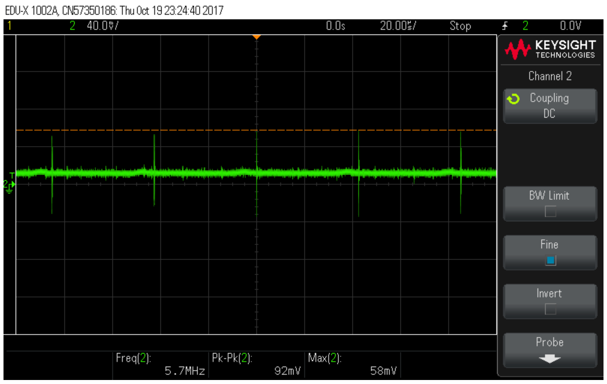
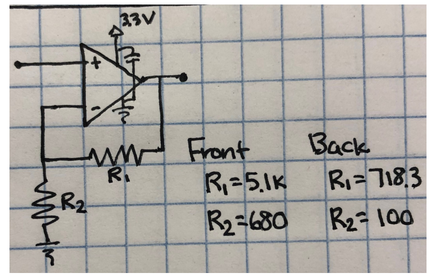


Figure 47: Peak Detector Output 4” away (back sensor)

Second Stage of Gain

The second stage of gain was introduced simply because the amplitude of the peak detector output was not large enough for the Uno microprocessor, ideally greater than 1V. This stage of gain is slightly different for the front and back sensors, like the peak detector. We used these particular components so that output of both sensors was relatively the same.



Front Track Wire

$R_1 = 5.1k\Omega \qquad R_2 = 680\Omega \qquad G = 1 + R_1/R_2 = 8.5$

Back Track Wire

$R_1 = 7718.3k\Omega \qquad R_2 = 100\Omega \qquad G = 1 + R_1/R_2 = 8.183$

At 1" away from the track wire, the resulting waveform for the back sensor has an amplitude of about 1.05V. At 2" inch away, the amplitude is about 640mV. At 4" inch away, the amplitude is about 150mV. The resulting scope traces is shown below.

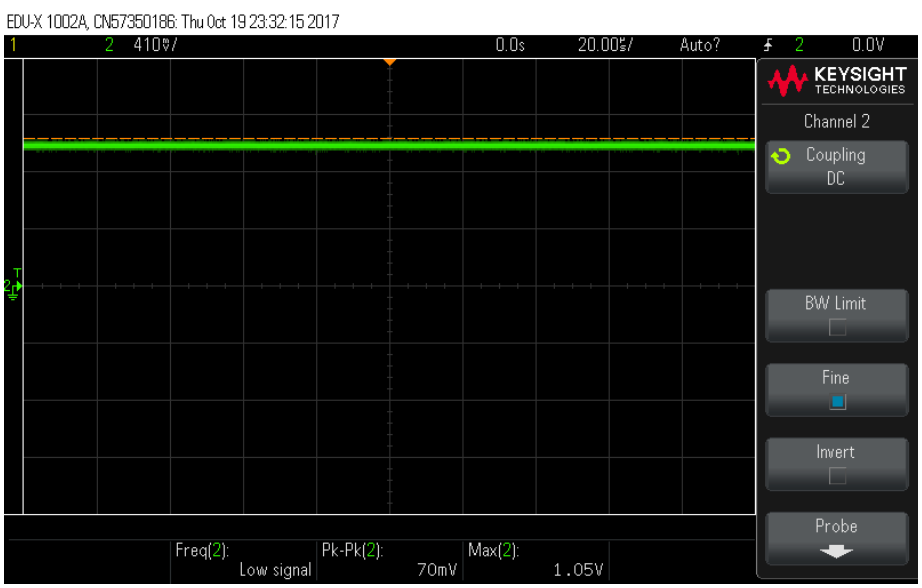


Figure 48: Second Gain Stage Output 1" away (back sensor)

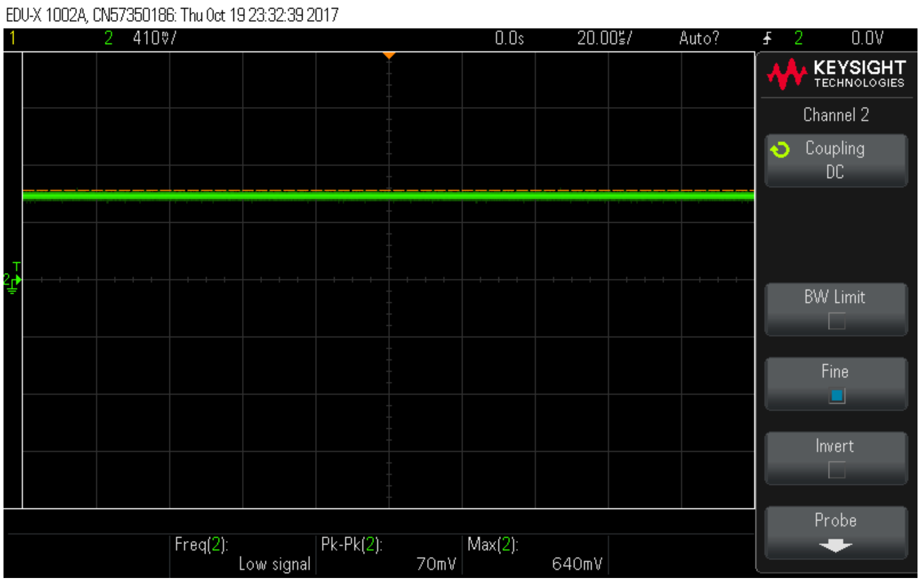


Figure 49: Second Gain Stage Output 2” away (back sensor)

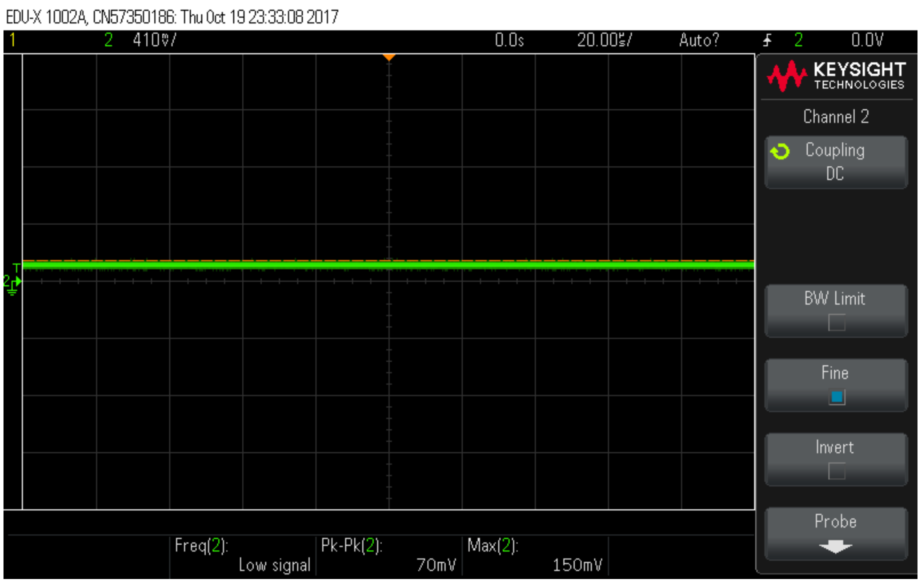


Figure 50: Second Gain Stage Output 4” away (back sensor)

Soldering the Track Wire Sensor

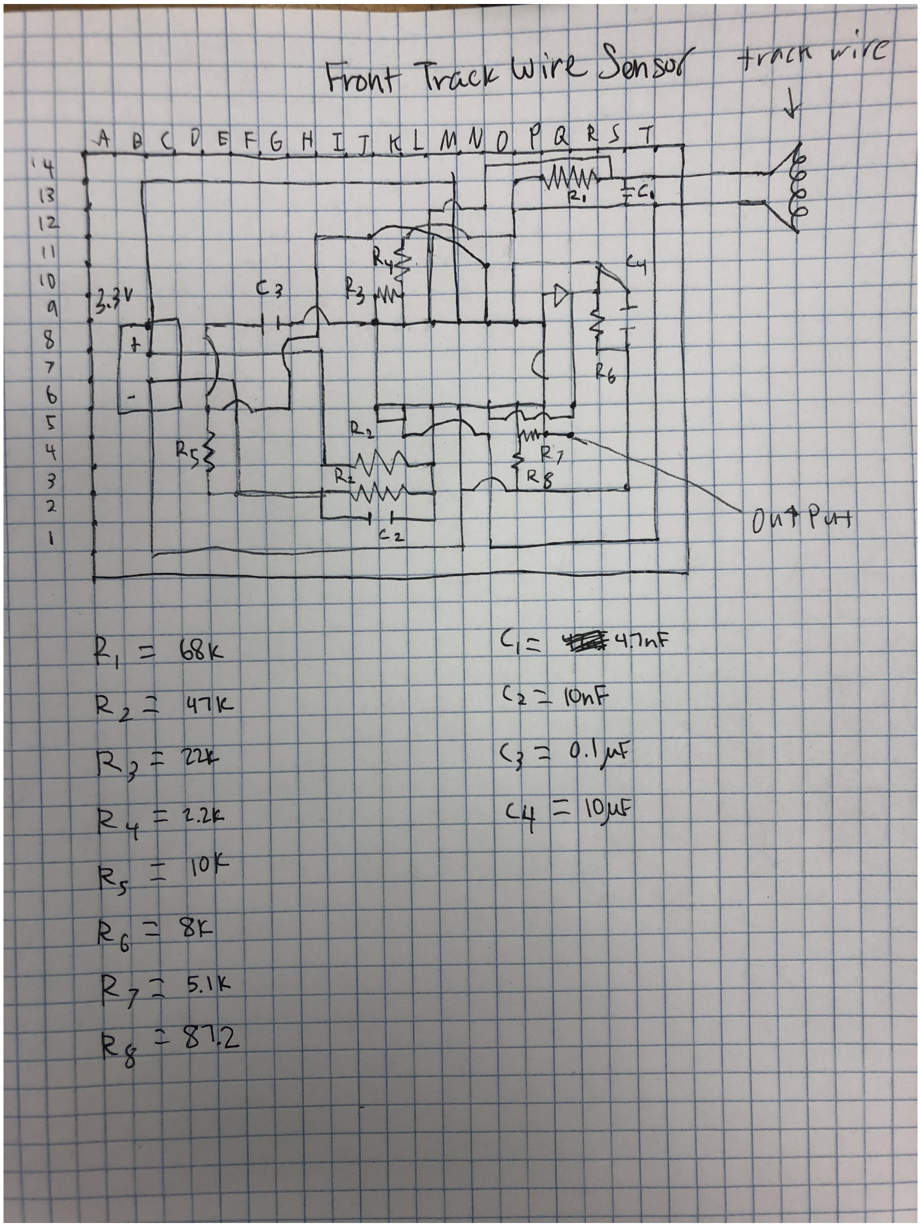


Figure 51: Perfboard Layout of the Front Track Wire

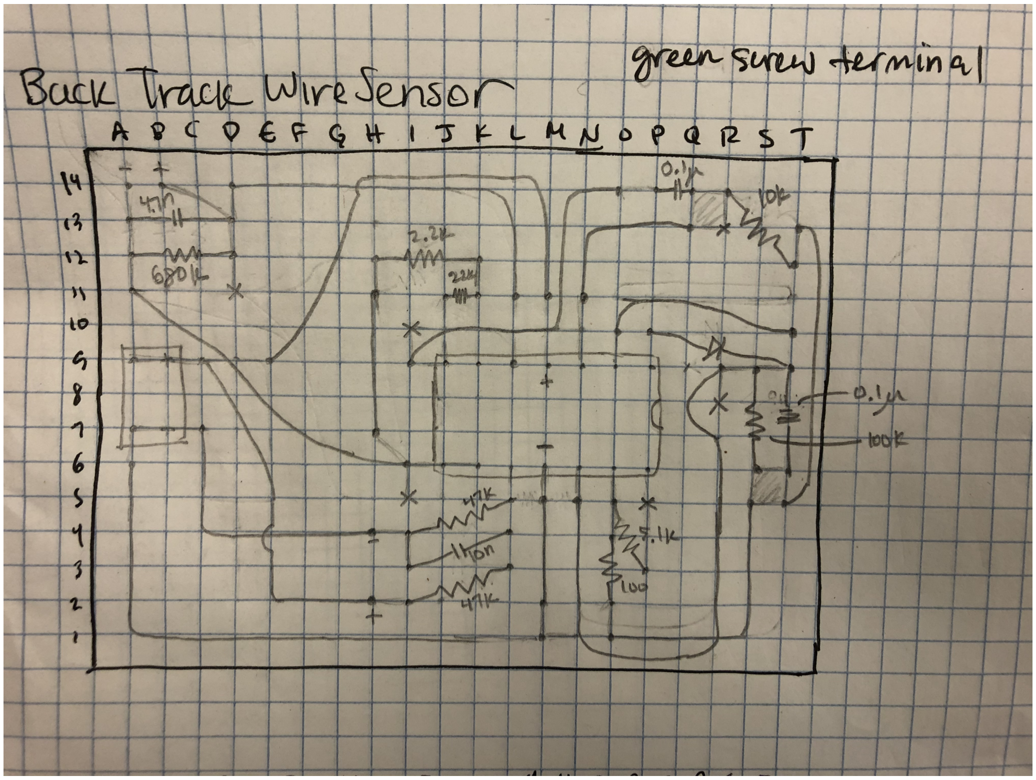


Figure 52: Perfboard Layout of the Back Track Wire

We used the same soldering method as the beacon detector for the track wire sensor. Using the perfboard planning layouts above, we constructed each component one-by-one, testing along for verification. One feature to note is that the inductors of the tank circuits were connected to the perfboard via jumper cables. This gave us more flexibility in mounting the inductors relative to the perfboards.

3.2.3 Bumper Sensors

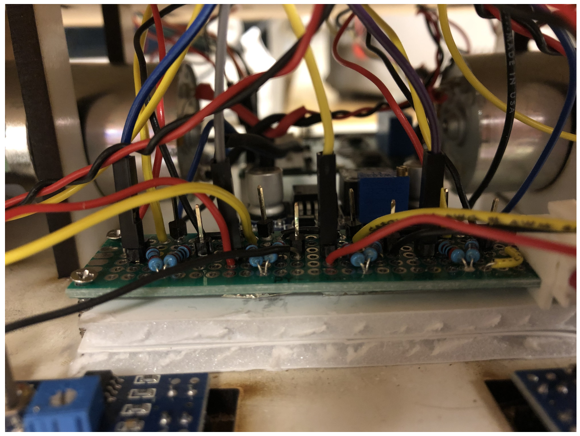


Figure 53: Bumper Breakout Board
Side view of the bumper breakout board on the bottom layer of the chassis

For our four bumper sensors on the bottom layer of the chassis, we chose simple SPDT switches (single pole, dual throw). This type of switch has three terminals: common, normally open, and normally closed. The switch connects the common terminal to normally closed, and when thrown, it connects common to normally open. We used the following circuit to drive a signal that varies from 0 to 3.3V depending on whether or not the bumper is pressed.

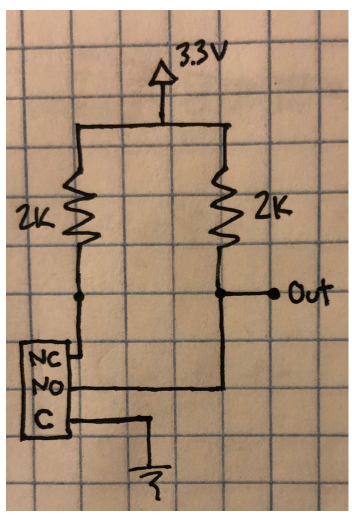


Figure 54: Bumper Switch Circuit Diagram

When the switch is open, there is no current flow, resulting in a 3.3V output. When the switch is closed, the output will instead read 0V. Since we use four different bumpers, we decided to create a bumper breakout board, which consist of power and ground rails and the four pull-up resistor circuits. This allowed us to only need one power connection for all four bumpers. To connect the bumpers to the board, we soldered wires to the bumper pins and soldered the other ends to the board. We originally placed male header pins at the output for jumper cables, but we were afraid that these connections would hold up under harsh maneuvers. The perfboard layout is shown below.

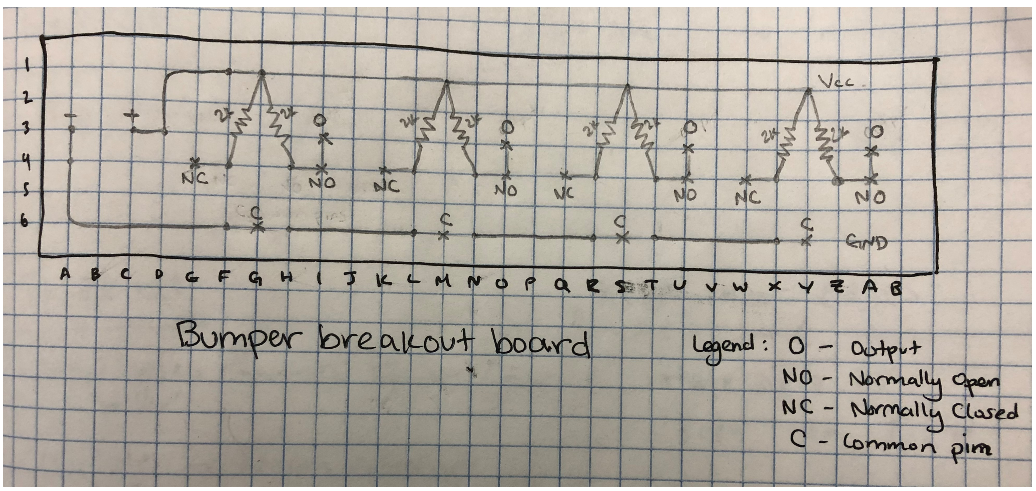


Figure 55: Perfboard Layout of the Bumper Breakout Board

3.2.4 Tape Detectors

As mentioned in detail prior in the mechanical design section, we opted to purchase pre-built tape detectors. Despite the circuit being relatively simple, we decided that the time saved creating and testing 6 identical tape detectors was worth the relatively low cost of purchasing them in bulk.

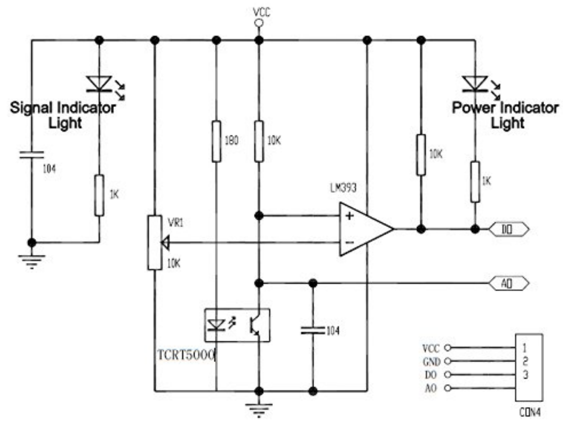


Figure 56: OSOYOO Tape Detector
Schematic of the tape detector circuit

3.3 Power Regulation & Distribution

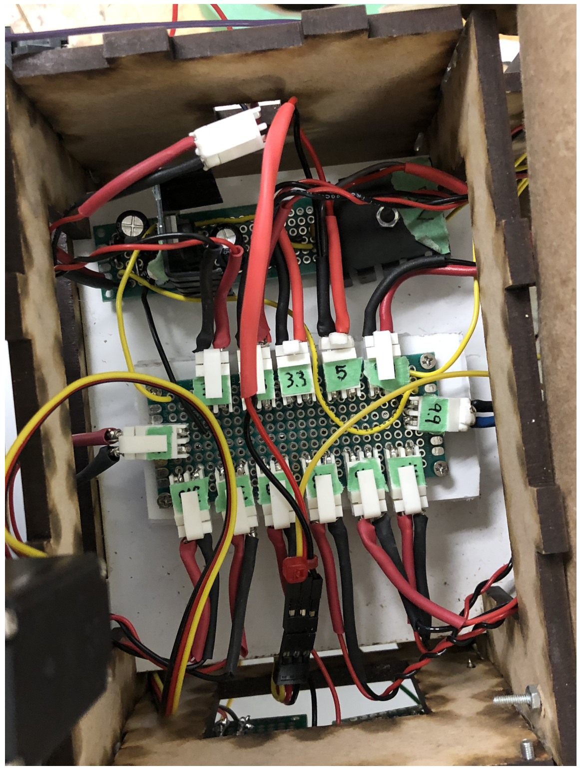


Figure 57: Top view of power regulation systems
Top view of power regulation boards inside the central stack.

To provide power to all the different components of our robot, we decided to construct a power distribution system, which included a power regulation board and a power breakout board. In the figure above, the smaller board on top is the regulation board, while the larger one on the bottom is the breakout board. These two board work together to provide three different voltages to GoldServo: 3.3V, 5V, and 9.9V. All the sensors (beacon detector, track wire sensors, bumper sensors, and tape detectors) require 3.3V. This is due to the Uno32 only being able to receive a maximum 3.3V signal. The ping pong ball servo takes in 5V, while Uno32 and the buck-boost converter, which powers the h-bridge, receive 9.9V straight from the provided battery.

Power Regulation Board

This particular board takes in 9.9V from the battery and creates two regulated voltages of 3.3V and 5V. This was accomplished with a LD33V 3.3V voltage regulator and a 7805 5V voltage regulator, each connected to a heat sink. The following circuit was used for each voltage output.

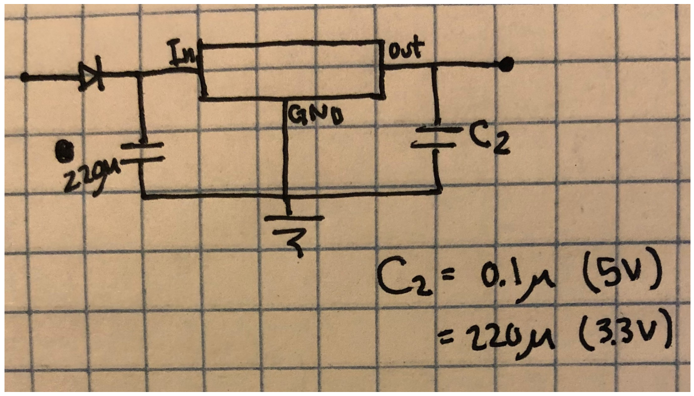


Figure 58: Power Regulation Circuit Diagram

The original version of the power regulation board used the same 0.1μ output capacitor for both 3.3V and 5V. However, as we found out through testing the beacon detector, the 3.3V regulation circuit was generating a considerable amount of noise, which increased with the number of sensors connected the power supply. This was due to the output capacitor not being large enough. Switching to the 220μ effectively filtered all the noise. The final perfboard design is shown below.

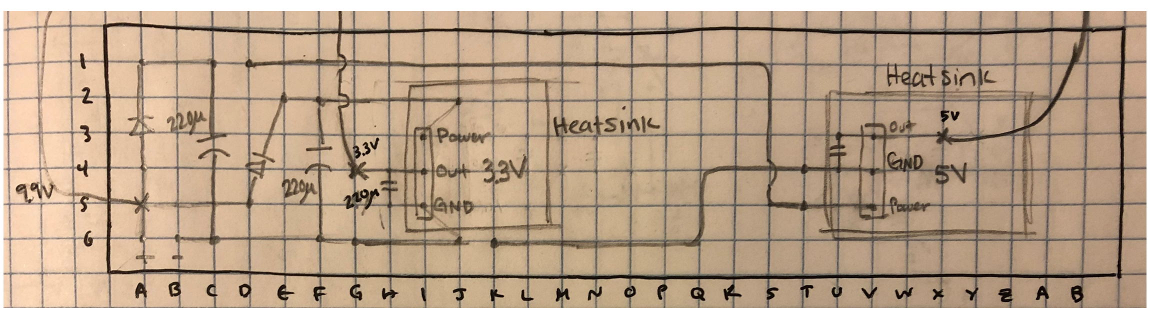


Figure 59: Perfboard Layout of the Power Regulation Board

Power Breakout Board

We were originally planning on using to male header pins for voltage outputs, but quickly abandoned this idea as we were afraid jumper cable connections would not endure harsh operating conditions. Thus, we switched to board-mounted rectangular connectors with female and male headers that ensure strong connections. Unfortunately, the female connector pins were not the right size for the given perfboard. To connect them to the board, we had to cut off the pins and solder short pieces stripped-wires to the ends. These wires were then soldered into the appropriate holes so that connector would lie flush to the board.

The breakout board consisted of ten 3.3V connectors, one 5V connector, and two 9.9V connectors. One of the provided perfboards was the perfect size to line the outer edges with evenly-spaced rectangular connectors. This made for easy power rail placement: the ground rail runs down the center of the board and the power rail lines the outer edge. For secure connections, we soldered in wires from the output of the power regulation board for the 3.3V, 5V, and 9.9V supplies to easily accessible holes on the breakout board. The final perfboard design is shown below.

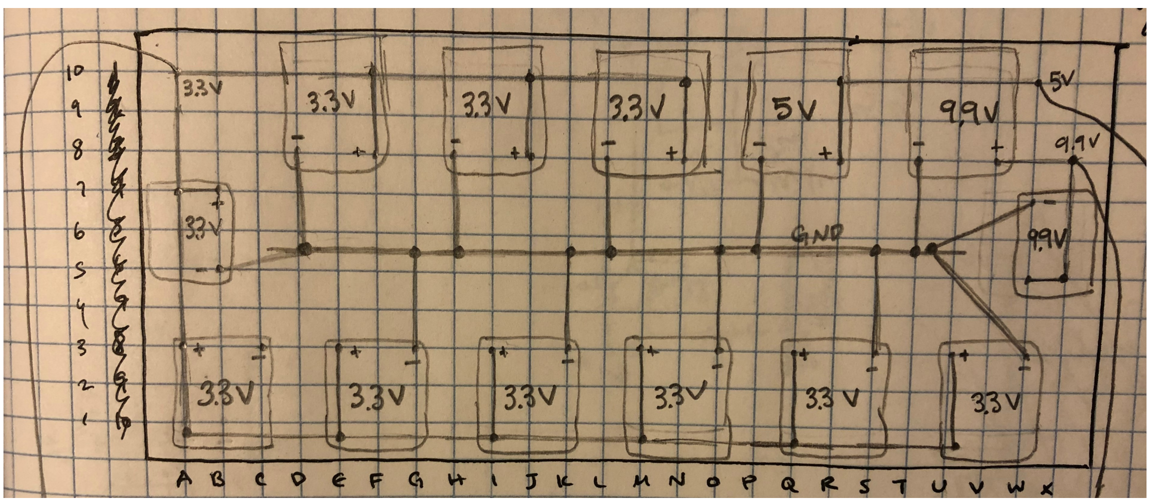


Figure 60: Perfboard Layout of the Power Breakout Board

4 Electrical and Mechanical System Integration

While mechanical, electrical and hardware driving and reading software was able to be developed in parallel, the integration of the physical components of the bot into an integrated package ready for programming was a critical path that needed to be completed before behavioral software could realistically begin development. We initially projected this integration phase to take a maximum of 2 days, which eventually proved to be over ambitious given the work that ultimately needed to be done. While purchasing of sensors and actuators was done well ahead of time to prevent shipping delays, acquisition of the uncommon M2 mini screws used to mount the bumpers delayed progress by 2 days. Additionally, we did not account for the time needed gauging and soldering up the various lengths of power ground and signal wire bundles for each sensor in our sensor suite and the motors and motor drivers. Although cutouts in the base plate were projected to be used to route all necessary cables and reduce wire clutter, the positioning of certain parts ultimately made this unfeasible, and only some cabling was routed trough these holes.

5 Software Design

We began work on the software towards the tail end of the project, and so had mostly full knowledge of the physical design while planning. We began work with the event checkers, leaving in the event checker macro framework so that we could test them independently later. We then built the framework of the whole state machine, including services for all sub state machines, before working on coding any of the particular states. While our states changed several times over the course of the project, we kept the number and placement of the sub-state machines consistent, which meant we never had to roll back any of this preliminary work.

Our process was to continuously test each part in chronological order before adding any subsequent parts. This meant we were continuously accounting for the physical handling of our robot, and that we had an understanding of how consistent our starting position would be on the next parts.

5.1 Hardware Drivers & Interfacing

5.2 Event Checkers & Services

Our final list of event checkers and their corresponding events are as follows:

- Bumpers (Press and de-press events for each bumper)

- Bottom tape sensors (on and off events for each tape sensor)
- Ball tape sensors (on and off events for the left and right tape sensors)
- Beacon Detector (found beacon and lost beacon events)
- Front Track Wire (found track wire and lost track wire events)
- Back Track Wire (found track wire and lost track wire events)

All of our event checkers employed a consensus threshold, which required the value to remain constant for a certain number of clock cycles before an event would be registered. We divided the event checkers in this way so that each event checker could have a different consensus threshold, which we tuned several times over the course of the project. Additionally, our track wires were read as analog values, and we employed digital hysteresis to determine when to push events. The front and back track wires each had different hysteresis bounds, which we tuned and re-tuned several times.

5.3 State Machines

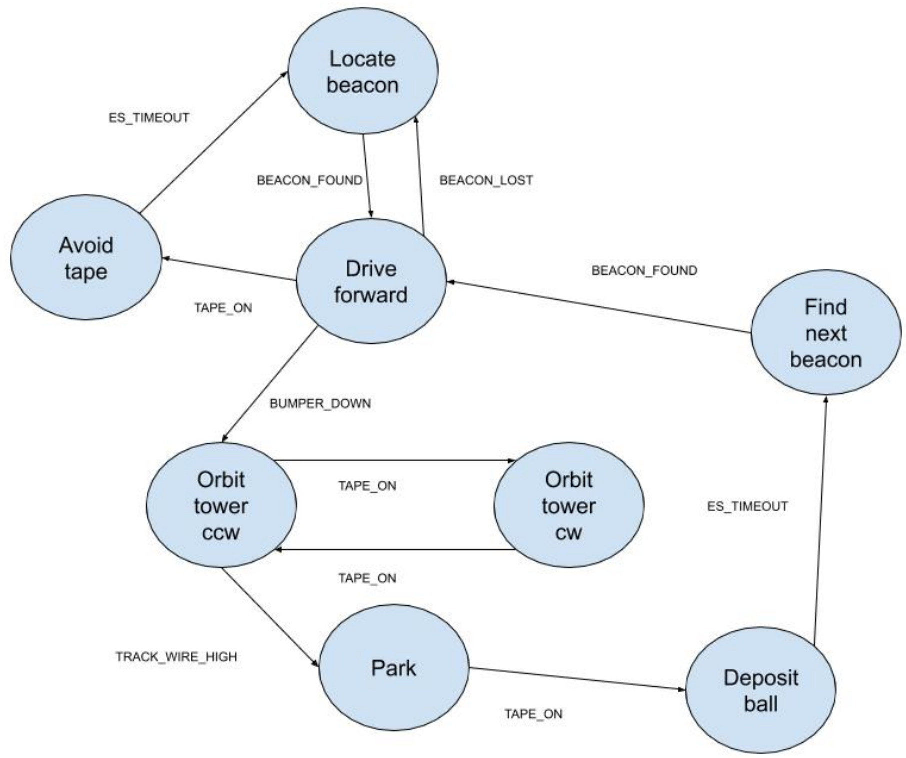


Figure 61: General State Machine Diagram
A rough picture of our Top-Level State machine diagram, which became more specific over the course of the project

Throughout the project we kept the above diagram relatively consistent, mostly adding additional states around these big ones to aid in transitions. We began by tank turning in place until we located the beacon, which ensured that we would always begin facing into the field, with no danger of running off the edges. Once the beacon was located, we would drive forward until we lost the beacon, found tape, or got a bumper pressed event. Losing the beacon would send us back to the Locate Beacon state, quickly correcting our course before continuing forward. Because we began facing into the field, finding tape was almost guarantied to mean that we had hit the center square. As such we forwent tape following in favor of a hand-coded maneuver to circumvent the center square (Avoid Tape state). If we got a bumper event, we assumed we had hit the tower and went into an orbit state.

The orbit state would take us around the tower in a counter-clockwise circle until we found tape or a track wire signal. If we found tape we would begin orbiting clockwise, taking us back around the tower while keeping the ball launcher on the correct side. We went through two methods of orbiting over the course of this project. Originally, we planned to rotate the beacon 90 degrees on its servo, so it would be facing the tower while orbiting. We could then use the beacon signal to correct our course as we held a constant distance from the tower. While this was supposed to make our orbiting more consistent, in practice it proved finicky, since the beacon's range was so wide at close distances. We instead opted for traditional wall-following, using a series of pivot turns guided by the left-side bumpers. This proved much more consistent, and was the version used in our final design. Once we found a track wire signal, we aligned with the tower and drove forward until we found the hole with the tape. After that we had a sequence of hand-coded motions to deposit the ball in the hole, move away from the tower and seek the next beacon.

5.3.1 Top Level State Machine

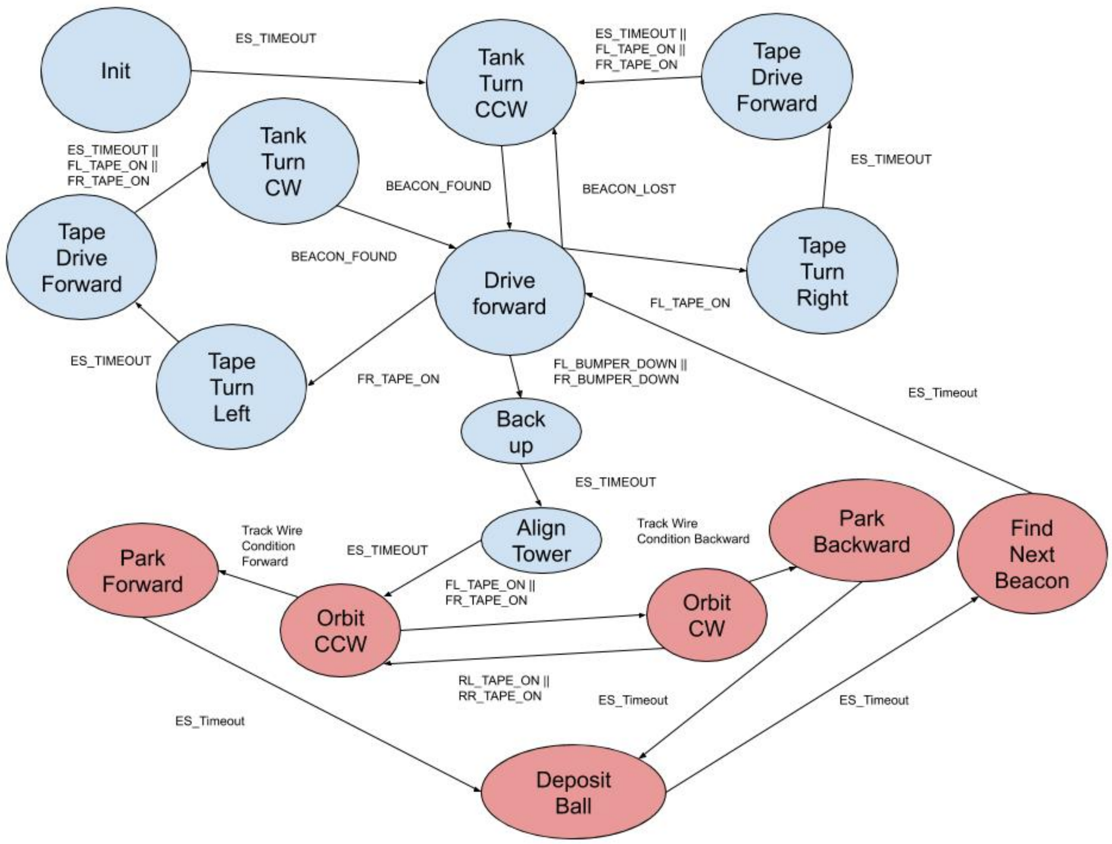


Figure 62: Top Level State Machine Diagram

The Full schematic for our top level State Machine. Sub-State Machines are highlighted in red

Our final Top-Level State Machine is similar in structure to the General schematic I described above, but with several additional small states to aid in alignment.At the very beginning we have an Init state that does nothing for a short period of time. This is to clear any events that happen when the robot is turned on for the first time. Instead of a single "Avoid Tape" state, we have a left and right path, depending on which tape sensor was tripped. In either event, we tank turn away from the tape, drive forward for approximately two feet, then enter the attempt to find the beacon again. There are now two different states for finding the beacon, one which tank turns clockwise and one which tank turns counter-clockwise. After avoiding tape, we enter the one that turns towards the tape we avoided, cutting down on time spent tank turning. Additionally,each Tape Drive state will cut early if it finds tape, preventing the robot from driving out of the arena while in this state.

On contact with the tower, we have a short sequence of backing up and tank turning before beginning the orbit. We then have different park sub-state machines for clockwise and counter-clockwise orbit. Both go into the same Deposit Ball sub-state machine, which goes into Find Next Beacon as before. The condition to go from Orbit to Park is that a Front Track-wire High event must be received, followed by a Back Track-wire High event, with no Track-wire Low events in between. The order of front and back is reversed if the robot is orbiting backwards. This ensures that the robot will be fully on the correct face of the tower when the transition happens, and minimises the chance of a false positive on one of the other faces.

5.3.2 Sub-State Machines

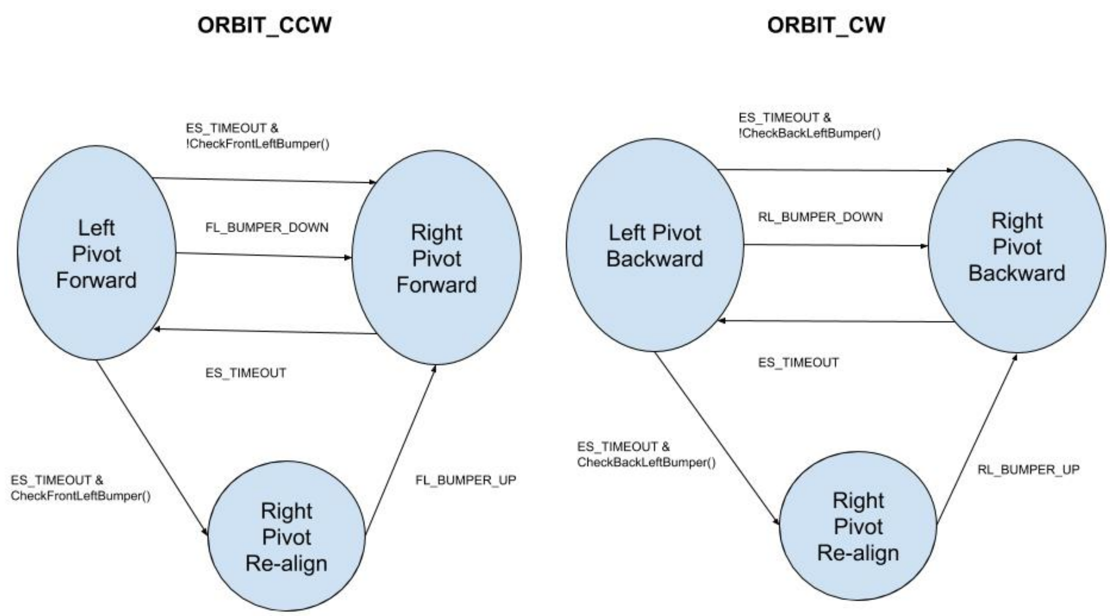


Figure 63: Orbit Sub-State Machines
Orbit Sub-State Machines, Counter-Clockwise (Left) and Clockwise (Right)

The Orbit Sub-State Machines operate by an alternate series of left and right pivot turns. Because both orbits keep the left side of the robot facing towards the tower, a left pivot turn is towards the tower and a right pivot turn is away from the tower. The orbit procedure is to do a left pivot until a bumper is pressed, and then right pivot for a short fixed amount of time.

While orbiting, there is the possibility that the robot gets stuck in a position where it is making a left pivot for a very long time. There are two reasons this can occur, and a timeout event in Left Pivot handles each differently. Firstly, it could not successfully de-press the bumper in the time it is in Right Pivot, and thus be waiting for a Bumper Down event while the bumper is already pressed. In this case, the state machine will see that the bumper is already pressed and go to Right Pivot Re-align, where it will turn until it gets a Bumper Up event. If the bumper is not pressed, it is likely that the robot is trying to round one of the tower corners, which can take some time. In this case, it would not be desirable to go to re-align, since the bumper is already de-pressed and no Bumper Event would ever come. In this case, we just go to the normal Right Pivot to un-stick the robot from the tower and allow it to round the corner faster.

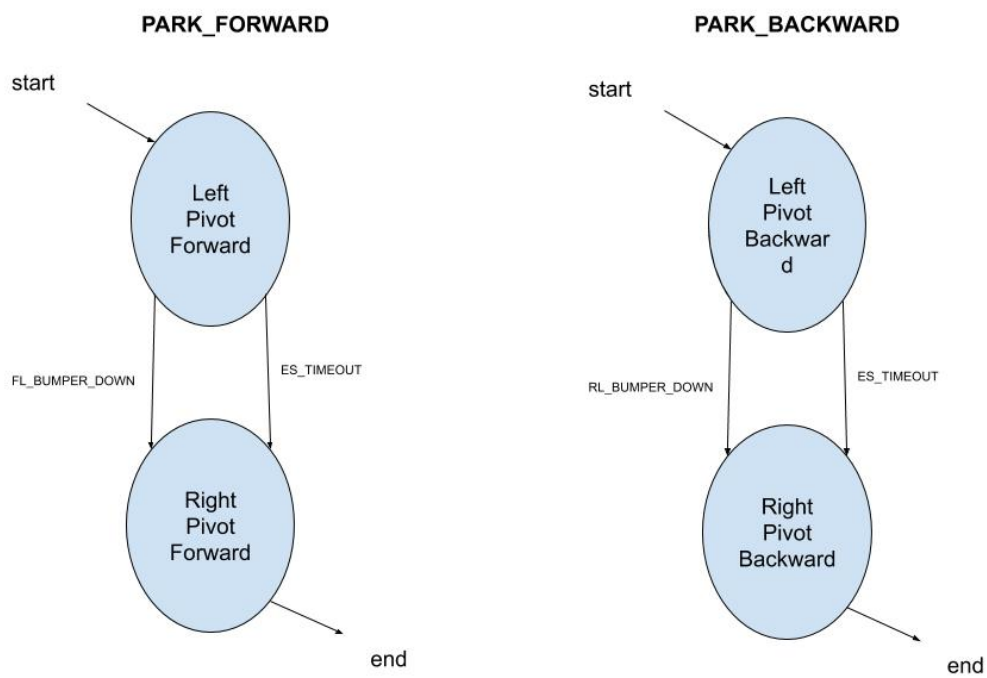


Figure 64: Park Sub-State Machines
Park Sub-State Machines, Forward (Left) and Backward (Right)

The Park Sub-State Machine is essentially one cycle of the co responding Orbit Sub-State Machine. This ensures that no matter which sub-state the track wire was found in, the robot will always enter Deposit Ball after a Right Pivot. This is important because the right pivot brings the robot parallel with the beacon wall, ensuring it can move along the beacon face smoothly. It also serves to bring the robot further onto the beacon face if it detects the track wire while rounding a corner.

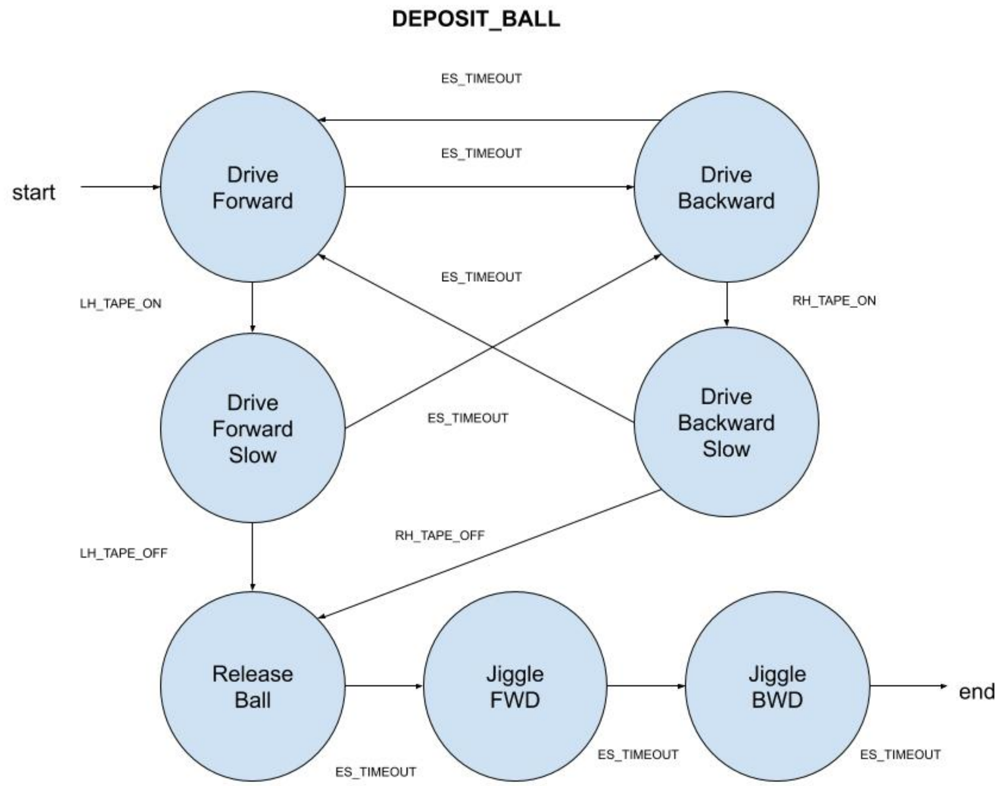


Figure 65: Deposit Ball Sub-State Machine

The goal of Deposit Ball is to find the hole with the tape on it and deposit a ball in said hole. It begins by driving forward until it gets a tape reading on the side tape sensor, or it gets a timeout event. The timeout is in case the robot started ahead of the correct hole, and will put the robot in a Drive Backward state. This Drive Backward state will also go until it finds tape or hits gets timeout (longer this time, to compensate for it having to go the whole length of the beacon). Once the robot has found tape, it must make sure the tape is actually tape and not the edge of the beacon tower (which also registers as a Tape On event). It does this by driving forward slowly until it gets a Tape Off Event, or it hits a timeout. A timeout means it has gone off the edge of the beacon, and must return to Drive Backward.

A Tape Off event means the tape was real, and it can safely go into Release Ball. An equivalent of Drive Forward Slow exists for when tape is detected while going backwards.

Once the correct hole has been found, the robot stops and releases a ball by moving the servo to the downward position. It will then do two short movements, forward and backward. This is to attempt to push the ball into the hole if the robot is not perfectly aligned with it. After this, the Deposit Ball routine is finished and the top level state machine will move on, after resetting the ball servo position.

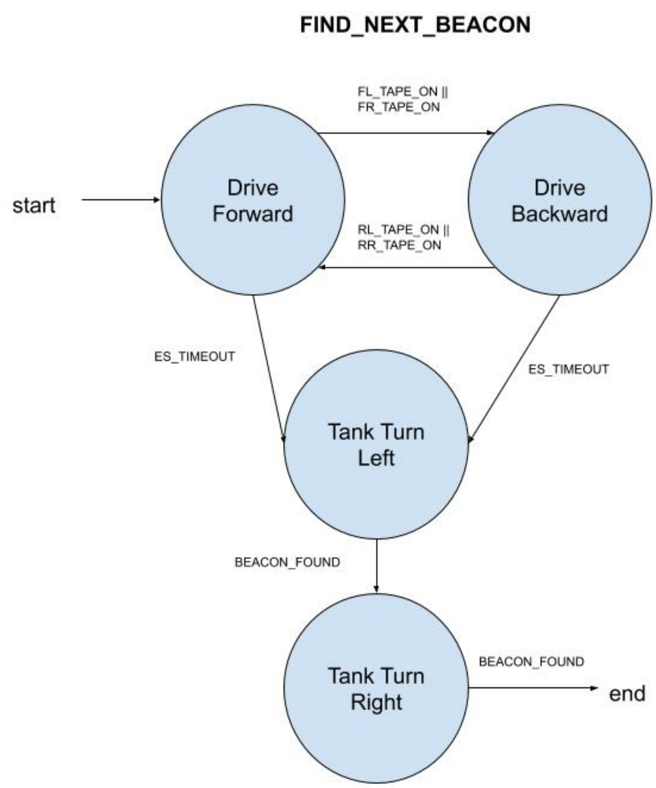


Figure 66: Find Next Beacon Sub-State Machine

The last Sub-State Machine is Find Next Beacon. This begins by attempting to get away from the tower. The robot does this by driving forward for a fixed amount of time, and if tape is encountered driving backwards for twice that much time. This positions the robot parallel to the beacon face but a full beacon length away from it.

It then tank turns left until it gets a Beacon Found, positioning the robot directly facing the near edge of the beacon. From here, the robot tank turns right, sweeping the whole field such that it will encounter the beacon it just deposited a ball in last. Once a beacon is found, the top level state machine can return to Drive Forward.

This process increases the likely hood that the robot will not return to the same beacon twice in row. There are configurations where this will not guarantee it, for instance if the previous beacon is between the robot and all other beacons (in the beacons "shadow" so to speak). However, in practice the robot will find a different beacon in most configurations.

6 System Testing

System testing was done by breaking down the overall objective of the bot into various behavioral milestones. This made it easier for us to focus on finding and handling edge cases within each major state machine of the overall HSM, preventing issues or edge cases early on from compounding as the state machine became more and more complex. The major milestones were broken down into:

- Initializing hardware and begin moving
- Detecting and reacting to bottom tape sensor events, enabling the bot to stay within the field
- Detecting a beacon and adjusting trajectory to approach the beacon accordingly
- Detecting a bumper event and transitioning into the initial tower circling position
- Orbit the current tower face by continuously bumping into the wall and adjusting heading accordingly
- Turning to next wall of tower once previous wall has been fully traversed without trackwire detection occurring
- Detecting trackwire presence an transition to finding the correct hole
- Aligning with the correct hole once side tape event occurs, then depositing a ball with reliable accuracy

- Departing from current tower and orienting GoldServo in order to find a different tower than the one just visited
- Tuning of case\tower setup related issues\errors

Initially, many of the early milestones such as basic movement we a simple matter of enabling state transitions to allow movement or tuning of the tape sensors to adjust to location dependant light levels. Locating the beacon signals introduced some road bumps, as we encountered unreliable response to 2KHz signals by our beacon detector, ultimately cause by a 1V noise source present on the 3.3V rail, mitigated heavily by the addition of a larger capacitor on the power boards. While hitting a tower and beginning to circumnavigate it was trivial, it was here we realized our enclosing orbit system would not be reliable after numerous trial runs and tuning of parameters, which is when we opted to make the switch to a wall hugging system instead. Detecting the track wires took a reduction in the gain of our circuits to work, as we found our bot was becoming confused by false positives due to too much range on the boards, detecting the trackwire through the walls of the tower despite being on other sides at times. Detecting, aligning and depositing a ball into the correct hole did not prove terribly difficult, requiring mostly calibration of tape sensors, modification of movement speed\duration, as well as fine tuning of the barrel length and angle. The method of departing the tower was a product of mostly trial and error to find the most consistent method, as one way of departing did not prove 100% reliable in all configurations of towers tested.

The edge cases mainly encountered with the finalized system mainly revolved around tight runs near the tower corners causing components to catch and torque and eventually break off. This was mostly rectified via removal of sharp angles on the bumper edges and form factor reduction of the bumpers themselves, although we found that catching a small portion of the bumper on the corner of the tower proved beneficial later, allowing GoldServo to "drift" around the tower corners slowly to ensure a very tight turn into the next face. Overall issues related to edge cases relating to major navigation behavior required more fine tuning of timer values and motor speeds than actual redesign of major software components.

7 Results

After much time spent debugging and tuning the behavior of GoldServo, we were able to get a system that could always find and deposit balls into the two S.A.L.T towers, while being able to complete the full 3 tower course in under 90 seconds with relatively high consistency. While major issues were resolved, some small tower configuration based errors persisted, and we noticed that the bumper sensors would sometimes not trigger, although the test harness indicated such sensors were working fine. Unfortunately, our bumpers decided to once again fail on us during the competition despite indications of them working reliably earlier, leading to a disappointing first round elimination.

Conclusion

To sum things up, we designed, built and programmed and small robot to deposit ping pong balls into towers. While it took some trial and error, as well as a lot of lab time, we ultimately were able to get the bot working with minimal issues.

Appendix

Project Expenditures

Here is the list of project expenditures showing that we did not exceed the $150 budget. Omitted from this spreadsheet is the 2 sheets of MDF that we purchased to make all the necessary component cuts, which would add another $10 to the total expenditure amount.

Item	Quantity	Price/Unit	Description/Link	Total Price
Greartisan DC 12V 60RPM Gear Motor High Torque Electric Micro Speed Reduction Geared Motor Eccentric Output Shaft 37mm Diameter Gearbox	2	14.99	https://www.amazon.com/Greartisan-Electric-Reduction-Eccentric-Diameter/dp/B072N84JX7/ref=sr_1_1?keywords=greartisan+60rpm&qid=1573344470&sr=8-1	32.76
Blue Belly 76mm78a wheels	2	2.5	Bought from another team for negotiated price	5
Acrylic wheel-to-axle mounts	1	2	Bought from another team for negotiated price	2
Bumper sensors (5 count from other team)	1	0.35	https://www.amazon.com/gp/product/B073TYWX86/ref=ppx_od_dt_b_asin_title_s00?ie=UTF8&psc=1	1.75
OSOYOO 10PCS TCRT5000 Infrared Reflective IR Photoelectric Switch Barrier Line Track Sensor Module for Arduino Smart Car Robot with 5 8Pin Female to Female Jumper Wires	1	11.99	https://www.amazon.com/gp/product/B07C69N65P/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1	13.1
HiTec HS311 Servo	2	0	Donated	0
More Bumper sensors(25 ct.)	1	7.99	https://www.amazon.com/gp/product/B073TYWX86/ref=ppx_od_dt_b_asin_title_s00?ie=UTF8&psc=1	10.15
Ball bearing rollers	1	10.99	https://www.amazon.com/gp/product/B01BV9ZUSU/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1	10.15
Buck/Boost Converter	1	3	Bought from another team	3
Total Cost	NA	NA	All the stuff to make the bot go zoom	77.91

Figure 67: Spreadsheet of all items purchased for this project

Video Demonstrating Functionality

Please excuse my terrible cameraman skills, I did not realize how much I was waving the camera around during the video.

Click Here for the video