ECE 445 Senior Design Laboratory

Final Report

Laser/Voice Assisted Cat Toy

Team # 9

Paul Jablonski (pjj3)
Rahul Grover (rgrover4)
Yutong Gan (yutongg9)

TA

Rui Gong

December 11th, 2024

Abstract

This document describes the development of a sensor-assisted cat toy intended for real-world use through features such as object detection, treat dispensing, and general interactivity for the cat. This is achieved through a laser sensor, vibration sensor, and microphone, allowing the toy to gather data from its surroundings. Most importantly, ethical considerations and safety regulations are considered when analyzing noise, power safety, and the materials used in creating this toy to legitimize the toy for commercial use while also keeping its manufacturing cost at a reasonable price. Overall, the purpose of this toy is to improve upon the shortcomings of traditional toys while remaining engaging for the cat.

Table of Contents

1. Introduction

1.1 Problem

Modern cat toys have some systems for automatically moving around, but rarely use any sophisticated sensors. This is commonly seen in commercial toys like balls that roll around in random patterns[1]. However, these widespread and commercial systems could use some serious improvements as problems exist in longevity, noise generation, user friendliness, and an overall lack of interaction for the cats[2].

These toys typically bang into walls without any preventative systems in place, leading to potential damage not only to the toy itself but also to the pet owner's home - on top of being terribly loud[2]. This is significant as owners may need to replace their cat's toys far more than desired[2]. Additionally, the typical toy's constant speed and random directional movements diminish the quality of a cat's play experience, often resulting in toys being left under furniture or cabinets unnoticed. With their fixed and unexciting movement, cats may often stare or fear these toys rather than chase them down as they would a live animal[2]. Thus, given the importance of engaging play for a cat's health, owners are often burdened by the current market's lackluster and rudimentary options.

1.2 Solution

It's proposed that these problems be resolved through a mouse-like toy, which has been seen before, but is now refined with multiple more advanced systems. The primary sensors consist of a distance measuring laser, a vibrational sensor, and a microphone in tandem with a miniature speaker. The first two serve in engaging the cat and improving both longevity and noise generation, whereas the final sensor system vastly improves user interaction through various voice commands. More specifically, the laser can be utilized to avoid collisions and detect ahead of the mouse, sparking movement changes that are more realistic for an animal. The vibrational sensor furthermore detects when the toy has been caught, which either dispenses a treat or plays dead depending on the toy's state. Finally, the aforementioned voice commands allow for the user to locate the toy at any time, or manually activate the toy itself without a need for physical contact.

Physically speaking, the non-rolling shape of a mouse also allows for more rigid and controllable movement. This is important for stabilizing the toy's primary sensors and allows for greater reactivity to its environment and consequently less noisy behavior. The sensors as mentioned also are accompanied by several motorized and more physical systems. A moving tail is used to mimic the more excitatory behaviors of prey, making it more engaging than a typical toy's static tail. This is accompanied by faster motorized movements and more realistic movement states as is regulated by a microcontroller and stepper motor-driven wheel system. Further speaking, a latch can be controlled through a servo and used for dispensing treats from the back of the toy upon being caught. And finally, a rechargeable lithium ion battery is incorporated and regulated by a circuit protection system for easy re-usability.

1.3 Visual Aid

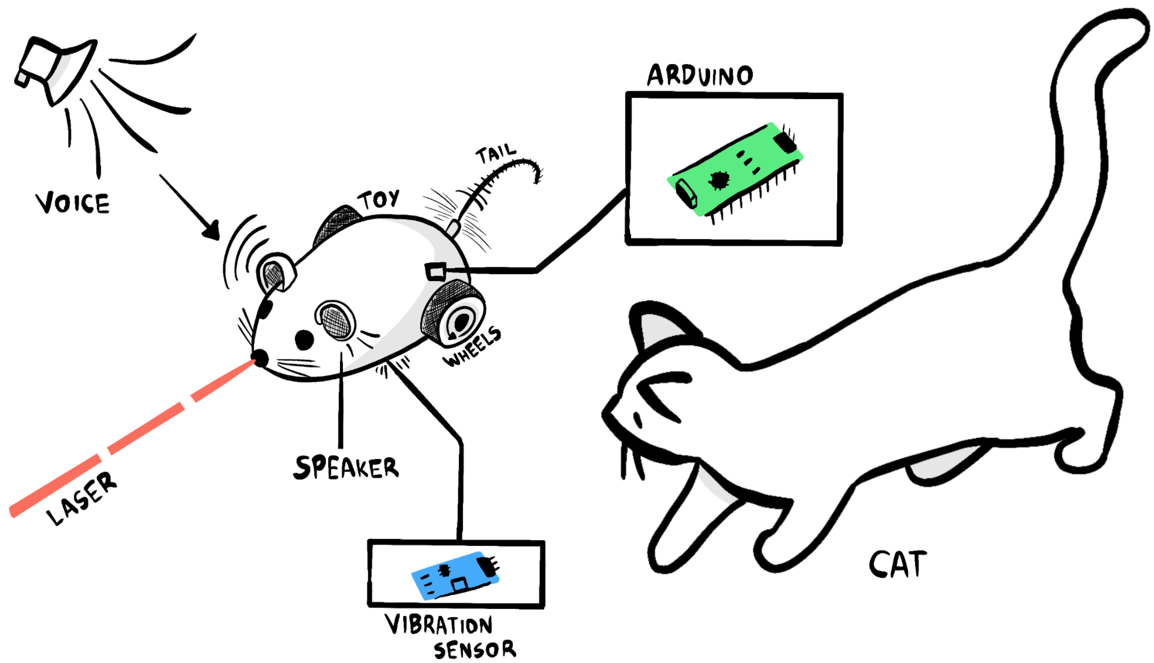

Figure 1. Concept sketch of cat chasing toy, laser distance measuring, and voice commands.

1.4 High Level Requirements List

In order to accurately claim that this project has been successful, certain high-level parameters and quantitative goals were initially set. These goals consist of the following:

- *Obstacle Detection and Avoidance* - The toy must be visually capable of stopping and turning if a wall or obstacle is detected ahead of it. It must then move in another direction continuing its play state while still avoiding other barriers. The detection range of obstacles should be observed up to one meter ahead, such that the toy may stop and perform this behavior properly.

- *Noise Reduction* - The toy must be quieter than the common and commercially automated cat toy. This is directly measurable in decibels (dB) and can be evaluated through recording a set distance away from the toys while they are active. It was decided a difference of 10 dB from the default environment to when the toy is activated would satisfy this noise reduction. This requirement is additionally testable in regards to quantifying and comparing audible wall collisions between the toys - where this project's toy would have far less collisions and ideally none.

- *Interactivity and State Changing* - The toy must include at least four movement styles, which must vary from the constant speeds seen in commercial toys. Furthermore, the tail must be capable of moving during these movement states. This can be tested through going through each of the microcontroller states and observing the wheel speeds changing variably, wherein the tail should be moving as well.

- *Voice Commands* - The toy should be capable of detecting human speech and recognizing the various voice commands assigned to it. The commands should be functional from at least five meters away, and at least three different commands should be included. Ideally these commands serve to locate, start, and stop the toy respectively.

1.5 Subsystem Overview

The toy itself comprises four total subsystems: the sensors, microcontroller, motors/outputs, and power subsystems respectively. The firstmost consists of all input data taken into the toy via laser, vibrational, and sound sensors, whereas the motors & output subsystem includes the corresponding outputs to stepper motors, servos, and the toy's speaker. This is determined through the system program, encapsulated within the microcontroller subsystem. Altogether, each of these three subsystems are powered and regulated by the power subsystem - as will be discussed.

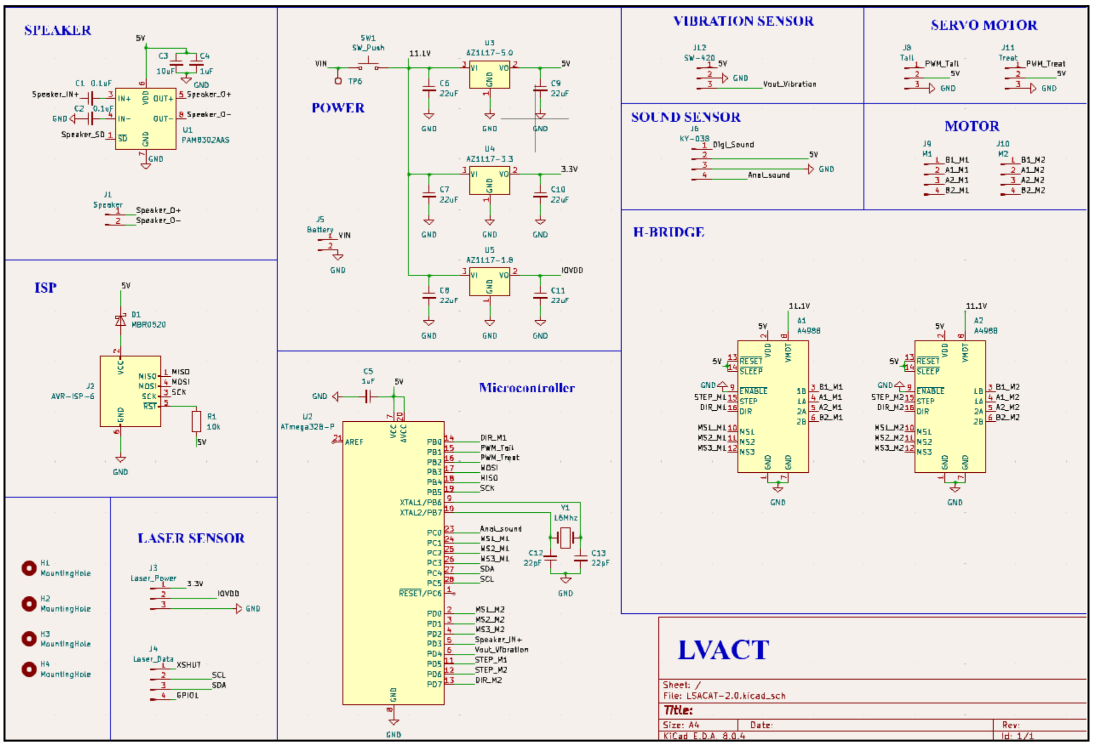

Figure 2. General overall circuit assembly including each subsystem's parts.

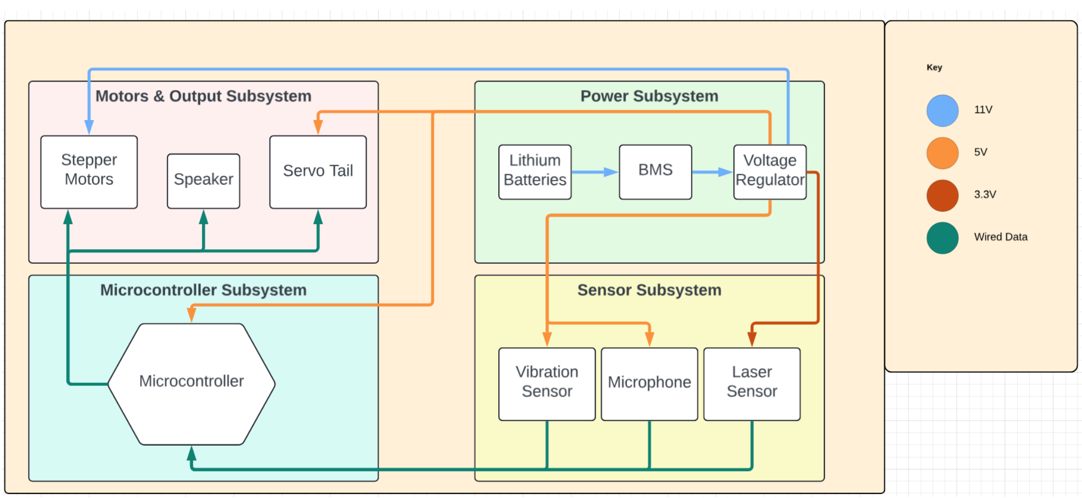

Figure 3. Block diagram representing each subsystem and their powered connections.

2. Design

2.1 Physical Design

The physical design process began with a smaller-frame model, which was based on the incorporation of DC motors as opposed to stepper motors. The shift from DC to steppers occurred however, due to the believed increase in precision that would be gained. As will be discussed in the toy's verification, this change was negative to the overall design, as the stepper motors were much more demanding power-wise and caused movement issues in the final product. Regardless, the following figure presents the initial render of the toy, which allowed for a compact design using DC motors as an alternative:

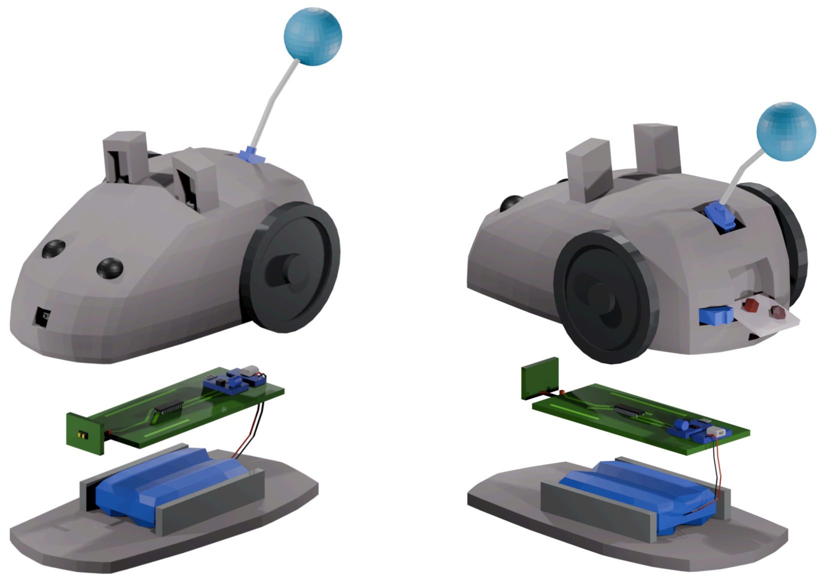

Figure 3. Initial concept render of disassembled front and back view of the toy.

In comparison, the final CAD physical design is presented below, which includes extrusions that were required by the stepper motors' inclusion, alongside the division of the product into assemblable parts:

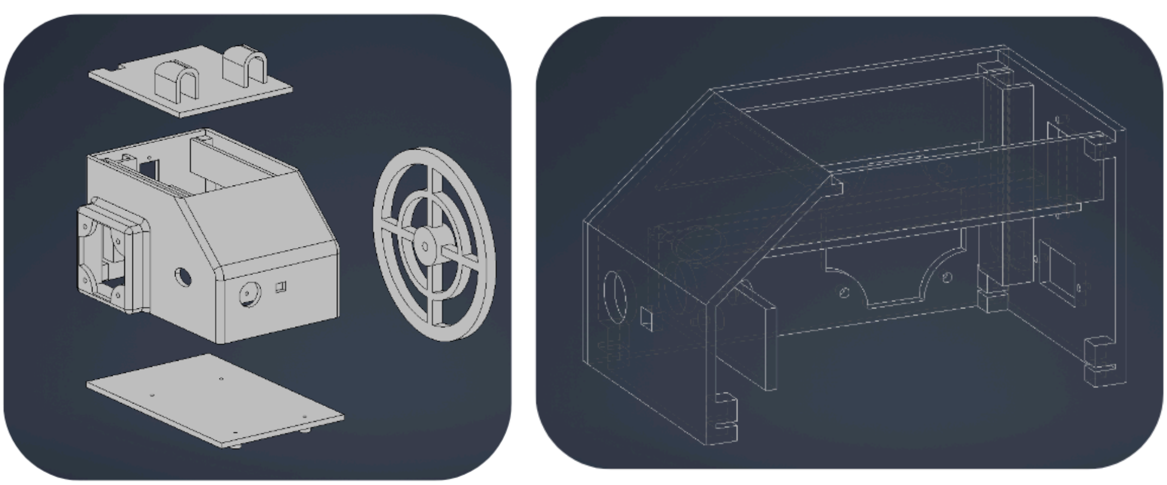

Figure 4. Finalized CAD model shown both externally and internally.

In the provided CAD figure, the toy is split into a main body, alongside a top cover, and bottom cover for mounting the PCB - which remained generally untouched going from the initial render to the final product. Mounting holes are incorporated for the sound sensor alongside the speaker on the sides of the toy. These were initially going to be placed in the ears of the toy, as shown in the figure that precedes this one, but the sensor chips were larger than anticipated. Thus, the holes were made in correction to this issue rather than finding and having to order smaller components.

Holes are additionally present for the laser sensor and button in the front. In an alternative design, the laser sensor bracket on the interior would have been moved closer to its front-facing hole, in order to avoid interference from its cone of vision. This issue was resolved during physical assembly however, as the laser was taped to the front panel rather than mounted in the bracket designed for it.

The back of the design features mounting capabilities for the tail servo on top, and the horizontal treat servo on the bottom. No major issues were present for these extrusions, and a large chute is featured near the treat servo's cover hole that is intended to allow treats to be loaded from the top cover of the toy.

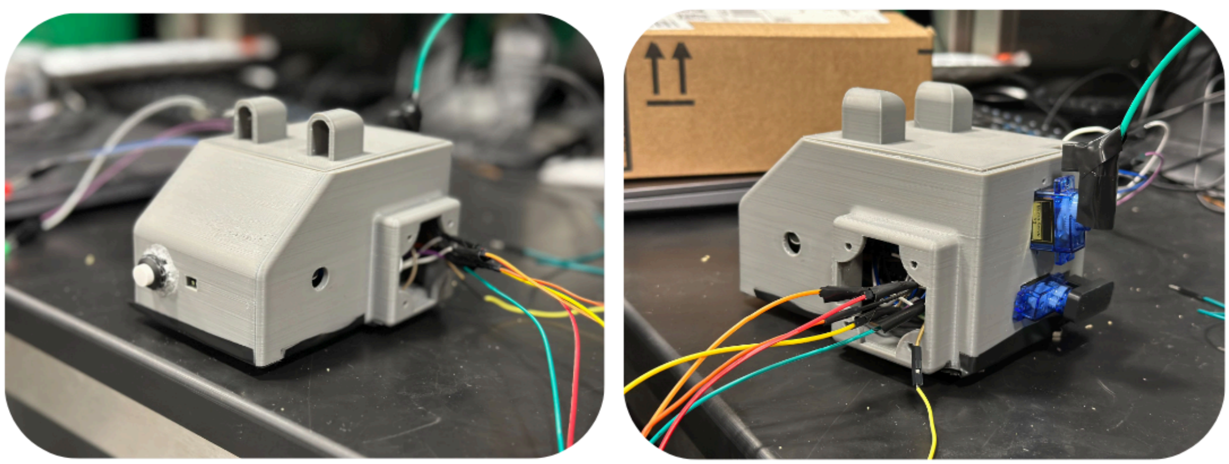

Figure 5. Assembly of final product.

The final aspect of the physical design process included assembly, which encountered some unforeseen problems, particularly in fitting the electrical components safely within the shell. Because of the internal wiring, keeping the shell shut was difficult, and after multiple re-assemblies, the bottom shell would no longer hold through friction as intended. Electrical tape was used to fix this on the bottom panel as printing a screw-based shell at this point would have taken too long.

Otherwise, super glue was used to fix non-vulnerable components such as the latch button in place, whose drying process was accelerated using baking soda, hence the opaque appearance around it. Internal circuitry such as the laser sensor were mounted to the shell using electric tape, which was more adjustable and safe to the design's functionality.

2.2 Program Design

The program design was generally modeled after a state machine diagram formulated during the early stages of this toy's design process. This diagram included the following general states: Start, Stop, Locate, Dispense, Rotate, and Movement. The transitions and interaction between states is pictured below:

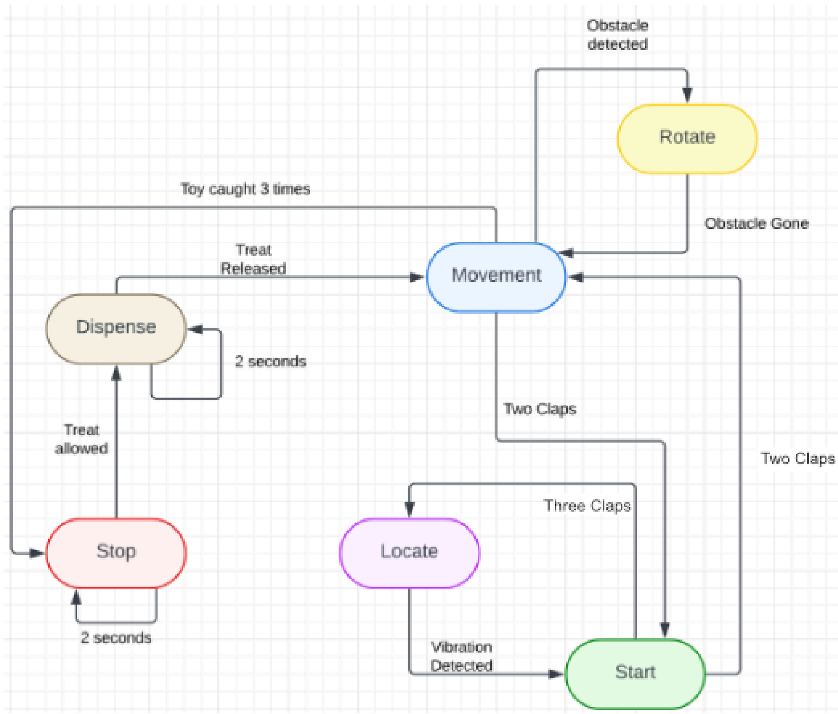

Figure 6. State machine diagram for program behavior.

Evidently, the clap system present in this diagram's transitions may raise questions, but the short answer is that the voice command system was substituted by one that focuses on large amplitude spikes. In part, this was due to the sound sensor, the KY-038, not being capable of performing frequency analysis - something paramount to voice detection and commands. It functions by polling for claps constantly in a non-blocking manner to the remainder of the code, and determines the claps by 40 dB volume changes occurring in short spans. Based on the amount of claps recorded two seconds after the last clap occurred, a command is executed according to the current state and diagram shown above.

```
void getSound() {
  currentValue = analogRead(Sound_Out);
  // Check if the change is sharp (value can be adjusted)
  if (abs(currentValue - previousValue) > 40 && millis() - timeClaps > 25) {
    numClaps += 1;
    if (millis() - timeClaps >= 2000 && (numClaps == 1 || numClaps == 2)) {
      numClaps = 1;
    }
    timeClaps = millis();
  }
```

Figure 7. Non-blocking clap system implementation.

From the figure above, a simple “if” statement proceeds for any usage of the function call, wherein the number of claps detected can be used to perform any action in the current state.

Two claps in the initial “Start” state begins the operation of the toy’s movement state, which is internally composed of three separate movement states as fulfills the high level requirements mentioned before. These different movement states consist of a standard forward movement, an accelerated forward movement, and a randomly rotating and accelerating movement. Two claps in movement can return the program to its initial state, whereas three catches triggered by vibrations can place it in its “Stop” and consequently “Dispense” state for rewarding the cat. Laser detection is active during movement as well, and similarly will move the toy into a stand-still rotation state until nothing is detected a meter ahead of it.

Finally, the initial state includes a three clap transition to the “Locate” state, which plays a repeating sweep tone until a pick-up or vibration is detected.

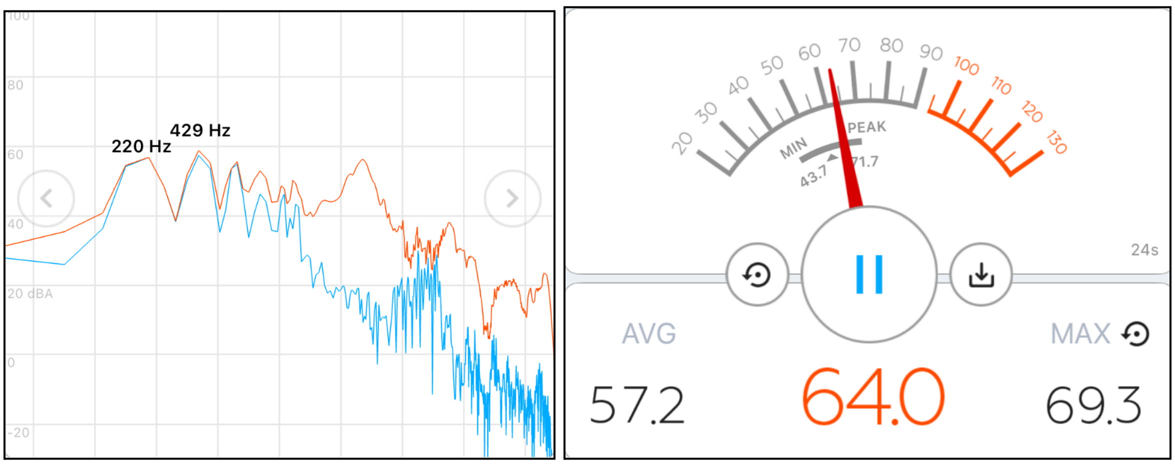

Figure 8. Frequency analysis of a noisy room alongside dB measurement.

The adjustment from voice to clap dependency in the commands system is justified in both its simplicity, but also effectiveness. In the original idea of using frequencies to decipher human speech, far more complex noise isolation algorithms would need to be implemented in the Arduino environment, something potentially not capable for the smaller ATMega328P microcontroller.

In the figure above, the clap design was capable of performing consistently under such noisy conditions, whereas speech recognition may be extremely inconsistent. The design choice of using sequential claps and not just one also eliminated the possibility of unwanted noise changes impacting the state transitions.

2.3 Electrical Design

2.3.1 Sensor Subsystem (Laser, Vibration, Microphone)

The sensor subsystem is capable of receiving data from the toy's environment in order to supply it directly to the microcontroller subsystem, which will decide upon motor and sound outputs. These sensors are vital for driving the toy's general reactivity to its environment. If the sensors fail to supply the microcontroller at the specified frequencies, then the toy will collide into walls, not change states upon being caught, and it won't be responsive to voice commands.

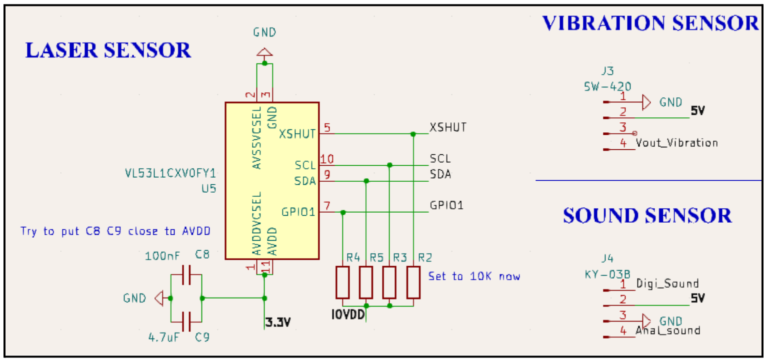

Figure 9. Sensor subsystem circuit schematic.

The sensor subsystem consists of a VL53L1X laser sensor, a SW-420 vibration sensor, and a KY-038 sound sensor.

The VL53L1X laser sensor has a maximum 4 meter sensing range and fast ranging frequency up to 50 Hz. The laser sensor was mounted on the front side of the toy for front-facing sight. This sensor helps detect the obstacles around the toy and generates distance-based control signals to avoid collisions. The outputs of this are sent to the microcontroller, which determines what to do movement-wise depending on how far obstacles are found. As will be shown in the later design overview, the movement state of the toy moves to a rotation state if the set distance threshold of one meter is triggered at any point.

The SW-420 is a highly sensitive non-directional vibration sensor. The sensitivity is remedied simply by changing the resistance of the potentiometer. The output signal of this adjusted vibration sensor is passed to the microcontroller to determine whether the toy has been caught or not. This is utilized during movement, but also in the toy's locating and audio playing state to detect if it has been picked up by a user to cease tone generation.

The KY-038 is a similarly sensitive sound sensor. The sensitivity can also be adjusted by changing the resistance of the potentiometer. The sensor is able to detect the amplitude of sound signals fed to it, and then sends the amplitude reading to the microcontroller. The sensor is able to receive commands to turn the toy on or off, and can play a sweeping tone as well to help locate the toy.

2.3.2 Microcontroller Subsystem

The microcontroller subsystem is generally responsible for receiving inputs from the sensor subsystem, while providing outputs towards the motors and outputs subsystem. For instance, this system intakes laser distance measurements and outputs the correct behavior to each motor to avoid collisions. This includes telling the motors to turn the toy until nothing is detected ahead of it, upon which the toy may resume moving. Beyond the example provided, this system is also responsible for processing user commands, generating tones, and determining when the toy has been caught based on the vibration sensor's inputs.

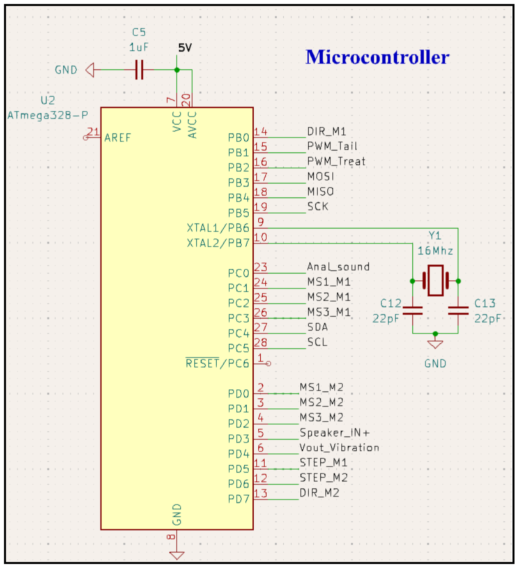

Figure 10. Microcontroller subsystem circuit schematic.

The microcontroller subsystem is driven by a miniature Arduino chip known as the ATmega328P. This chip is capable of handling up to 14 digital input/outputs and 6 analog inputs. PWM signals are also capable of being processed by 6 of the digital pins, which is vital for the motors and their driver inputs.

The sound from the KY-038 sound sensor is handled through analog inputs in order to preserve sound data, otherwise digital would treat it as binary.

The VL53L1X laser sensor communicates through I2C protocol through a digital pin in order to provide information to our program about object proximity.

The SW-420 vibration sensor is also handled through digital input as it contains a threshold trigger for activation. This was manually adjusted to determine the severity of impacts and if a cat has caught the toy.

A 16 MHz clock is additionally connected, which ensures the system has accurate clocking in the program-aspect of its design. This is important for the motors as the write speeds need to be accurate.

2.3.3 Motors & Output Subsystem (Wheels, Tail, Latch, Speaker)

The motors and output subsystem allows the toy to move forwards, backwards, and directionally across any surface. Based on what the sensors send to the microcontroller, different predefined movements will be performed by the motors. This includes avoiding obstacles and engaging in different play styles through various set speeds. The tail also moves variably based on the microcontroller's current state to randomize the toy's behavior further. The speaker system is also part of this subsystem, as it outputs sounds when prompted by the microcontroller's processing of commands.

The motors & output subsystem consists of two Nema 17 short body motors, two SG90 servo motors, two A4988 stepper drivers, a PAM8302 sound amplifier, and one JSM 2.5mm speaker.

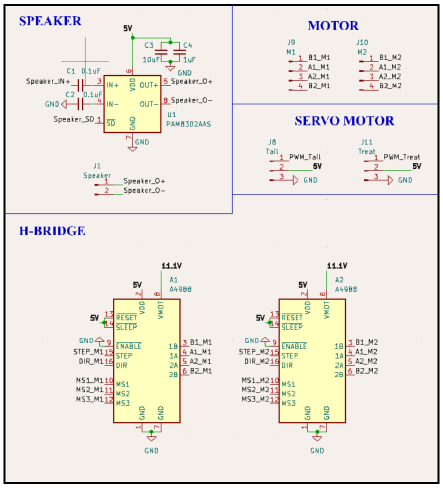

Figure 11. Motors & outputs subsystem circuit schematic.

The Nema 17 stepper motors are compact yet powerful motors suitable for precise movement. They are used to control the primary motion of the toy, allowing for smooth and controlled movement patterns that engage the pet in playful interaction. These motors are driven by the A4988 stepper drivers, which provide adjustable current control and thermal protection to ensure stable and efficient operation.

The SG90 is a servo motor that can be driven by 4.8V to 6V. Based on the write input, the motor is used to mimic the behavior of tails, and is further used to dispense treats after the toy is caught. The treat dispensing operates through covering a small hole full of treats, which is then uncovered by the servo.

The PAM8302 sound amplifier is a low-power audio amplifier designed to drive small speakers. It enhances the audio signals generated by the toy, ensuring clear and audible sound effects for the user.

The JSM 2.5mm speaker is a miniature speaker component used to produce sound outputs, including the sweeping tone used for the audio location state.

2.3.4 Power Subsystem (Battery, BMS, Power Button)

The power subsystem provides power to the entire device, and it handles voltage regulation to every other subsystem. It utilizes multiple linear fixed regulators to supply different voltages to different sensors as required. This is seen as the laser sensor requires 3.3V, but the vibrational and microphone sensors need 5V. Furthermore, the overall power distribution is enabled and disabled through a latching power button.

The power subsystem consists of a 11.1V lithium-ion battery, three AZ1117 voltage regulators, and a CYT1091 button.

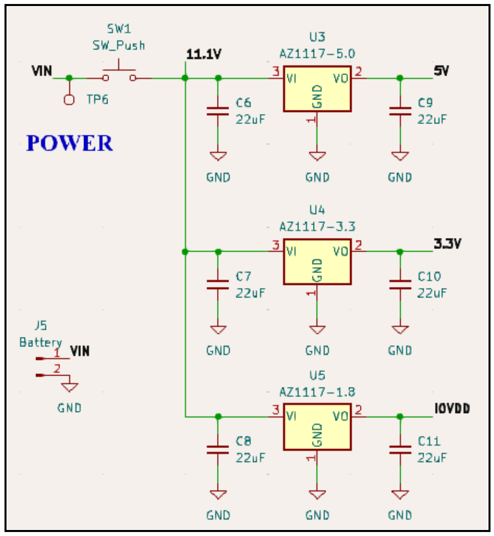

Figure 12. Power subsystem circuit schematic for battery input, voltage regulators, and toggleable button.

The 11.1V lithium-ion battery was thought to hold enough voltage and amperage to power all less demanding subsystems, while also driving the power-intensive stepper motors in the motors subsystem. With a capacity of 2600 mAh however, this was not true, as will be discussed in the quantitative findings later on.

The AZ1117 voltage regulator provides stable voltage based on the design's various needs. One AZ1117-5.0 is needed to provide a stable 5V to power the majority of sensors, the microcontroller, and the speaker. One AZ1117-3.3 and one AZ1117-1.8 are additionally utilized to provide 3.3V and 1.8V for the laser sensor.

A simple CYT1091 latching button toggles the toy's power supply based on whether the power connection is completed or not.

2.3.5 PCB Translation of Circuitry

The electrical design of this cat toy required designing a primary PCB that is capable of connecting all sensors as inputs to the microcontroller, and all motors and outputs as well. The following figure displays the translation of the prior circuit figures to a comprehensive PCB unit:

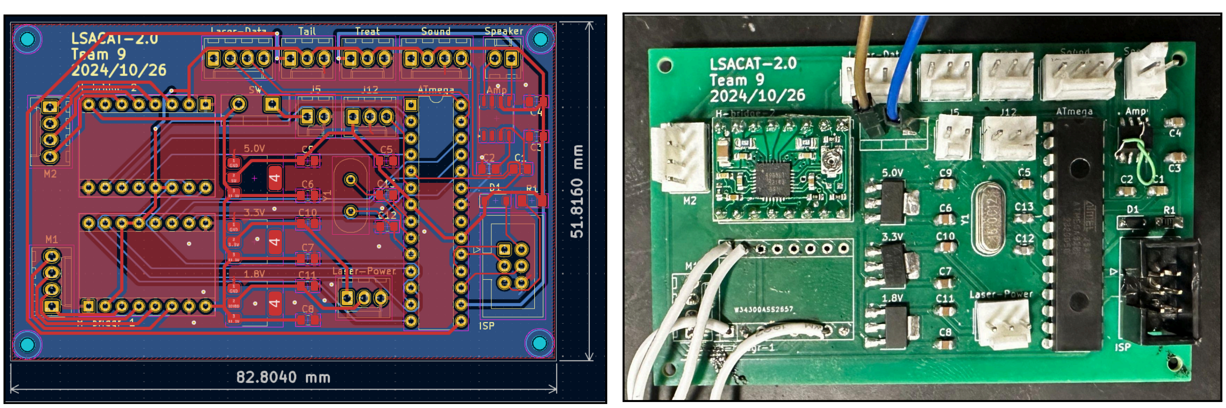

Figure 13. Digital main PCB design in comparison to soldered board.

This primary PCB contains all necessary connections for power management in addition to the inputs and outputs that were mentioned before and labeled in the left part of the figure. This is evident by connection J5, which contains the positive and negative leads for power input.

Most notably, the finalized PCB did not contain soldered connections for both stepper motor drivers, which was caused by the motors performing inconsistently. The white wiring in the bottom motor driver in the right part of the figure displays the alternative for displaying motor outputs: wired LEDs. The final demonstration and product simply represented the motor movements through LED representations of direction and stepping signals per motor. This was justified as the motors were incapable of switching directions or even providing enough torque to move the shell itself. Generally speaking, the amperage of the battery chosen was not enough to satisfy the 1A rated requirement by each motor. Having two motors and plenty of other components simultaneously only raised the power requirements of the cat toy, where this trade-off and exclusion had to be made. Based on testing with the in-lab power supply, the power needed to supply all sensors and the servos, excluding stepper motors, was around 1A on its own. With the 1A requirement per stepper motor, the total current required is around 3A which was not achievable.

Another PCB was going to be included in the design, which would have incorporated the VL53L1X laser sensor on a custom board. This board had minor issues because the laser sensor chosen sends signals at 1.8V whereas the microcontroller operates at 5V signals. This would require an additional bi-directional voltage shifter to the regulator incorporated in the design, thus a commercial substitute was used.

2.4 Updated Tolerance Analysis

The original design of this toy involved the tolerance analysis of its motor speeds and wheels in order to identify the optimal speed the cat toy should move at. This was done so the toy doesn't collide into walls and is programmed correctly. While the motor system does not function as planned, it's still important to outline this calculation for potential future work or for envisioning the motors as if they were working. The speed of any given wheel of radius 'r' millimeters with a motor rotating at a rate of 'n' revolutions per minute can be determined using the following equation:

$$velocity \ = \ \frac{n}{60} \times 2\pi r = \frac{n\pi r}{30} \ mm/s$$

For the newly updated Nema 17 motor the maximum RPM is rated at around 800 revolutions per minute assuming significant torque demands, which are present in this case. Meanwhile, the shell that was designed had a wheel size of 53.518 mm in order to allow for a caster wheel underneath the toy.

$$velocity \ = \ \frac{800}{60} \times 2\pi(53.518) \ = \ 4483.513 \ mm/s$$

This translates to an ideal velocity of 4.484 m/s, which is obviously not going to occur in realistic conditions due to drag, un-ideal wheel friction, and other environmental factors. But, this value may be utilized to determine the upper bounds of time the toy needs to reach a full stop. Using the chosen laser detection range of 1m for obstacles, some basic kinematics can be computed here to determine the necessary time for the toy to stop. A safe and realistic realistic linear deceleration rate of 3 m/s^2 has been arbitrarily chosen, as Nema 17 documentation does not provide such values, leaving the following:

$$v_f = v_i + a * t$$

$$\text{where } t = (v_i - v_f) / a = (4.484 - 0) / 3 = 1.495 \text{ seconds}$$

This means the toy has 1.495 seconds assuming such a high deceleration speed, which justifies part of the program design in the toy's final version. The "Stop" state was declared to stop the motor inputs for 2 seconds before dispensing a treat, which gives a half-second gap between this ideal calculated time for the toy to fully come to a stop.

3. Cost and Schedule

3.1 Bill of Materials

The overall cost of this project was determined by the individual parts and the required quantity per part. The following table summarizes these part costs followed by a brief labor analysis and additional costs for 3D printing the toy shell and ordering the PCB design.

Part Name & Number	Quantity Needed	Cost	Purchase Link
11.1V 2600mAh Li-ion Battery	1	1 x $45.42ea = $45.42	11.1V Battery
AZ1117 Voltage Regulator	3	3 x $0.38 = $1.14	Regulators
SW-420 Vibration Sensor	1	1 x $6.49 = $6.49 (5pk)	Vibr. Sensor
SG90 Servo Motor	2	2 x $6.99 = $13.98 (3pk)	Servos
KY-038 Sound Sensor	1	1 x $6.29 = $6.29 (5pk)	Sound Sensor
Nema 17 1A Short Body	2	2 x $11.99 = $23.98	Step Motors
Treedix JST 2.5mm Speakers	1	1 x $8.99 = $8.99 (4pk)	Speaker
PAM8302 Speaker Amp	1	1 x $9.95 = $9.95 (5pk)	Speaker Amp
ATmega328p Microcontroller	1	1 x $14.99 = $14.99 (2pk)	Arduino
A4988 Stepper Driver	2	1 x $7.99 = $7.99 (3pk)	Driver
VL53L1X Laser Sensor	1	1 x $17.99 = $17.99 (2pk)	Laser
CYT1091 Latching Button	1	1 x $6.99 = $6.99 (12pk)	Button
TOTAL	**N/A**	**$164.20**	**N/A**

Table 1. Costs and parts list.

The parts alone result in a cost of $164.20, which including a sales tax of approximately 9% results in a cost of $178.98.

Accounting for labor costs, it may be assumed a $48 hourly salary as the average in the United States for such engineering designs. In this project it's assumed three people are working on the project at a rate of about 8 hours each per week for 8 weeks. Therefore, there's a computed total labor cost of $9,216.

In terms of additional costs, the 3D printing of the shell cost around $500 as a personal 3D printer was purchased. Accompanied by another $25 for replacement parts, the costs are then totalled as such:

$178.98 + $9,216.00 + $30.00 + $500.00 + $25.00 = $9,949.98 total

3.2 Schedule

The following table contains the planned schedule for the weeks following design review and the work distribution alongside it.

Week	Overall Goals	Distribution
10/7	- Peer Review (10/8) - Design Review (10/9) - PCB Review (10/11)	1. Paul - Review microcontroller pins/start program 2. Yutong - PCB creation from circuit schematic 3. Rahul - Parts ordering and arrival estimation
10/14	- Team Evaluation - PCB Order Placement - Arduino Program Drafted	1. Paul - Full program outline completed 2. Yutong - PCB completed before order date 3. Rahul - Assist with PCB and analysis of it
10/21	- CAD Translation - Physical Model Planning	1. Paul - Begin moving 3D model to CAD 2. Yutong - Circuit placement planning for CAD 3. Rahul - Improve movement system in code
10/28	- Begin Parts Testing - Finish the Software - 3D Printing Prep	1. Paul - Finalize the software implementation 2. Yutong - Test arrived parts for functionality 3. Rahul - Research 3D print and best materials
11/4	- Toy Assembly - Shell Printing	1. Paul - Finish CAD model 2. Yutong - Assemble PCBs into toy's shell 3. Rahul - Get the model printed
11/11	- Revisions if Needed - Subsystems Testing	1. Paul - Code review if necessary 2. Yutong - Circuit review if necessary 3. Rahul - Subsystem testing/verification
11/18	- Mock Demo - Team Contract Fulfillment	1. Paul - Write team contract expectations/roles 2. Yutong - Prepare demo steps 3. Rahul - Write agenda/team issues
11/25	FALL BREAK	1. Paul - FALL BREAK 2. Yutong - FALL BREAK 3. Rahul - FALL BREAK
12/2	- Final Demo - Mock Presentation - Extra Credit Video	1. Paul - Presentation introduction/objective 2. Yutong - Presentation design slides (circuits) 3. Rahul - Presentation design slides (outputs)
12/9	- Final Presentation - Final Paper - Lab Checkout - Ensure Notebook Quality	1. Paul - Final report analysis/lab notebook check 2. Yutong - Final report drafting /lab notebook check 3. Rahul - Final report figures/lab notebook check

Table 2. Project schedule week by week.

4. Requirements & Verification

4.1 Requirements & Verification Results

Requirement	Verification Results
The toy should be able to detect any obstacles within the range of 1m and automatically move away from it to avoid collision.	Success - An obstacle was placed 1 meter in front of the toy, with consistently reproducible detection rates of > 95%, where the LEDs for direction motors were toggled as a result. A range of around ~20º was also identified for the operating width of the laser.
The toy should be able to identify if it is caught.	Success - The toy identified itself in a caught state following a light tap on top of the shell frame. Upon tapping 3 times, the toy changed to its dispensing state, where the treat servo rotated 45º and the direction and stepping LEDs were shut off until dispensing was completed.
The toy should be able to turn on/off through human speech. A recorded sound should be played after a “locate” command.	Majority Success - The toy could switch between movement (on) and the initial (off) state through a substituted clap-based system. Audio was also played with a ~15dB change to the toy’s environment during “locate”.

Table 3. Sensor subsystem requirements / verifications.

Requirement	Verification Results
Toy should begin in a state ready for user command activation. Should react to commands “start” and “locate”	Success - The toy evidently began in an initial state, where two claps triggered its movement operation as a “start” command, and three claps transitioned it to its “locate” command, playing ~15dB audio sweeps.
The toy should stop either when the vibration sensor detects the cat caught it, or the command “stop” is given	Success - The toy stopped and entered a dispense state upon being vibrated three times, which was evidenced by its LED shutoff and servo rotation. Two claps during movement would also revert it to its initial state.
The toy should only dispense a treat to the cat when the toy is caught successfully 3 times. Once a treat is dispensed, it should continue moving along.	Success - Once more, upon being tapped three times, the servo rotated to 45º for 2 seconds, and then rotated back, re-initiating the main motor LEDs to signify the beginning of movement once more.

Table 4. Microcontroller subsystem requirements / verifications.

Requirement	Verification Results
The toy should be able to move forward and backward at the average speed of 1m/s.	Partial Failure - While stepping speeds of ~590 steps per second were achieved digitally with LEDs, this was not translated to the motor to produce the 1m/s speed desired. This stepping speed required was calculated as: $SPS = (1m/s)/(0.339m) * 200$. The desired linear speed, divided by the wheel circumference, times the motor’s rated steps per revolution, determined the stepping speed needed evidently.

The toy should be able to completely stop at its maximum forward speed of in 1s.	Majority Success - Despite not moving, the toy was capable of transitioning to a stop state in any of the three movement states in its state transition diagram. This was done through two claps being detected, where the LEDs representing stepping speed would be disabled immediately.
The toy should be able to turn 90 degrees right or left.	Success - As the toy is capable of transitioning to the rotate state, as shown in figure 17, it is capable of rotating any amount of degrees with one direction LED indicator on, and one off. This was verified through rotating the toy 90^{o} upon detecting an object until no object was detected and rotation ceased.
The tail driven by the servo motor should be able to move up and down in different frequencies depending on states.	Success - The tail servo moved in a non-obstructive manner during all primary movement states with varying frequencies. We went through all three movement states and the rotate state, and recorded over 3 tail movement occurrences for each state - verifying its functionality.
The treat should be able to be dispensed after the toy is caught.	Success - As presented in the extra credit demonstration, upwards of three 15mm treat substitutes were able to be inserted in the treat hatch and dispensed through the treat servos removal of the cover upon state change. This dispense state change occurred following three catches in the toy's movement state.
The speaker should play a recorded sound properly upon command receival.	Success - Upon three claps in the initial state, the toy transitioned using its "locate" command, where a 500 - 1000 Hz sweep tone was played until the toy was tapped or picked up.

Table 5. Motors & output subsystem requirements / verifications.

Requirement	Verification Results
Must provide stable 5V, 3.3V and 1.8V voltage with 5% regulation.	Success - The 5V, 3.3V, and 1.8V linear voltage regulators were measured with 2%, 3%, and 4.8% error rates respectively, derived from figure 14 and shown mathematically below it.
The battery should be able to power the toy for at least 20 minutes.	Partial Failure - The 11.1V lithium ion battery was capable of supplying power to all systems except for the motors. This was short-lived however, as its charge had diminished to < 0.3 V over the course of testing. A 9V clip-based battery was connected and functioned, but assembly of the toy had ruined its ability to function, due to EMI or unintentional connections.
The button should be able to turn the toy on/off	Success - The power button successfully worked regardless of power source. Pressing the latch in would power the toy and boot it in its initial sound-recipient state, whereas depressing it returned its power supply to a measurable ~0V for all subsystems.

Table 6. Power subsystem requirements / verifications.

4.2 Other Quantitative Results

4.2.1 Power Subsystem Results

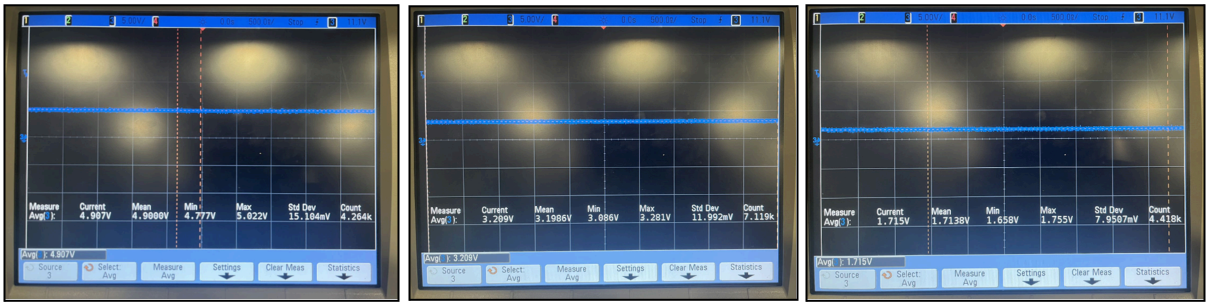

Figure 14. Tests results for 5.0V, 3.3V, and 1.8V regulators.

The figure above displays the output for the three different voltage regulators found within the power subsystem. The error for each was computable using the expected value and the test result values.

$Err_{5V} = (5.0 - 4.9) / 5 = 2\%$ $Err_{3.3V} = (3.3 - 3.2) / 3.3 = 3\%$ $Err_{1.8V} = (1.8 - 1.714) / 1.8 = 4.8\%$

All of them were within 5% regulation, which satisfied the verification requirements for voltage regulation. Other requirements such as the battery lasting 20 minutes were not met as the battery had been drained prior to demonstration and formal testing. The requirement regarding the power button's functionality was fulfilled however, as it powered the system on and off interchangeably.

4.2.2 Motors & Output Subsystem Results

Figure 15. Direction and step signal generation.

Figure 16. Motor input signal without and then with motor connected.

The first figure shows the creation of control signals and inputs for the motor system, which evidently works well for the direction and step signal generation. The problem occurs when a motor is connected, as seen in the next figure. The left-side is a clean step signal with 11V while the motor is disconnected. But, the right-side shows the same signal with the motor connected; the wave is deformed because when the motor is rotating, a huge ground noise will be generated and it will disturb the control signal.

Besides, the power source is simply not strong enough to power two motors at the same time, which makes the motor vibrate instead of rotate. Because of these problems, the motor was tested using LEDs, which will be shown in the next section. The other output here was the speaker, which verifiably generated a +15dB change to its environment, and held a sweep tone upon a locate command input.

4.2.3 Sensor & Microcontroller Subsystem Results

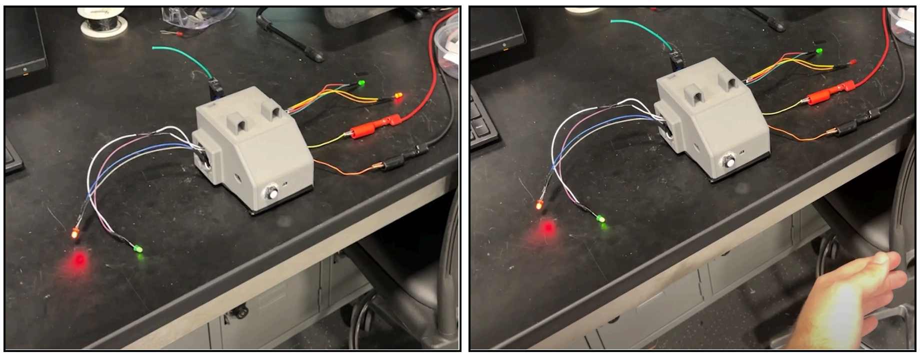

Figure 17. LED verification of laser sensor operation, red LED changes with detection.

The figure above shows the LED testing approach used to verify the remainder of this projects' requirements, which includes the display of motor stepping in green, direction changes based on microcontroller states in red, and general sensor reactions based on the current LED configuration. The test shown above in specific, is utilized to verify the operation of the laser sensor, which visibly changes the red LED to signify the turning of the toy.

Beyond this case, the LED tests were used to verify the state changing of the toy through clapping and sound sensor spike detection. The figure below presents and verifies the dB spike that occurs following the triggering of the audio-emitting "locate" state, which displays proper microcontroller transitioning:

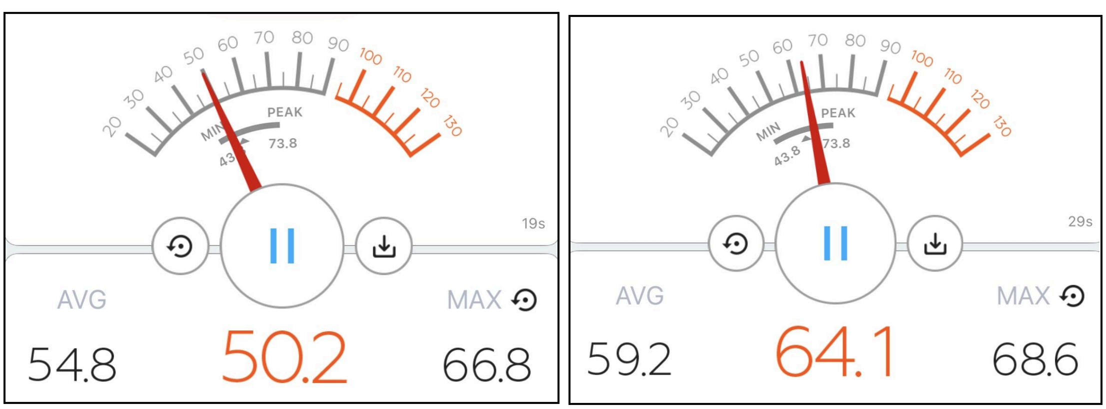

Figure 18. ~15 dB volume change measured from a meter away upon "locate" state change.

5. Conclusion

5.1 Accomplishments

Our toy was successful in aggregating the data from our sensors, supplying this data with proper voltage to the microcontroller, and finally performing logic on this data to properly handle outputs to our motor drivers for the wheels and our servo-motors for tail movement/treat dispensing. As discussed previously, our laser sensor detects objects within 1 meter, and triggers the rotate state in our state machine to force the toy to rotate until it is out of range of the obstacle. This is verified through the LEDs attached to our motor outputs when performing testing on our firmware. Additionally, our vibration sensor properly detects up to three interactions before triggering the servo motor to dispense a treat by moving from the hole covering them. Finally, our microphone sensor properly detects input from the user in the form of claps, and moves to the corresponding movement state (two claps), or begins playing a noise from our speaker (three claps) to allow the user to locate the toy. All of our components were soldered on our PCB with the firmware programmed into the toy through a 6-pin USB-ASP programmer. This was then placed into the shell of our toy on which we mounted our PCB, sensors, and motors.

5.2 Uncertainties

The major challenges we faced were related to our motors. The disturbance caused by the ground noise generated by the motors is too big for them to function properly. Consequently, the power supply is not stable enough which makes it impossible for us to control the speed of our motors precisely. We also ended up using a commercial module for our laser sensor because of the lack of a bi-directional voltage shifter. Finally, the relatively big size of the battery and stepper motors make our toy bigger than we initially expected, which we did not expect as we wanted our toy to be smaller.

5.3 Future Work/Alternatives

Some future work we identified for solving issues related to power instability would be to calculate the total power consumption with DC motors and see if they are able to function appropriately in comparison to the stepper motors. We did not use the stepping pins on our stepper motors in our final design so we believe that replacing them with DC motors would be a simple replacement that could also potentially solve the issues with ground noise. Additionally, we also believe that redesigning our PCB with more capacitors will shield noise from the motors and allow us to continue to use the stepper motors and develop our project further with more precise movement by then using the stepping pins. However, this would also require us to redesign the PCB and get a new order shipped. Finally, we believe that using a 10-pin USBASP programmer with TSX and RSX pins will be useful for viewing console outputs when debugging the state machine. We previously used LEDs to debug our firmware as a result which led to the development process taking a lot longer than expected since identifying the bugs was extremely difficult.

5.4 Ethical Considerations & Safety

With regards to analyzing this project ethically, our analysis is primarily following the IEEE Code of Ethics in order to ensure our design process went smoothly[3]. We also identified some codes from the ACM Code of Ethics in order to complement the IEEE codes, but also to provide guidelines for us in technical considerations[4]. Our foci revolve around upholding these specific codes, and the following sub-codes:

IEEE 1-5 To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors.

As we are doing a semester-long design project, we had to note all problems and errors so we did not repeat mistakes, but also so we made our project the best it could be. Our time and budget was limited, so being open to acknowledging our errors helped us get past them sooner. This involved collecting and carefully monitoring real data during our development process. Furthermore, we kept team member contributions clearly indicated so we could refer to them for help when errors arose, but also so we could give credit where credit was due.

IEEE 2 To treat all persons fairly and with respect, to not engage in harassment or discrimination…

We also ensured each teammate felt welcomed with their ideas, and that we treated each other properly. All decisions were discussed carefully and equally through either online meetings, text conversation, or in person. We made a group chat and did well at coordinating any clarifications or changes we planned on making. This involved confirming with each member that our suggestions are okay and that they don't disrespect each other's work. This avoided hiccups in our project and helped us work more efficiently.

ACM 2-1 Strive to achieve high quality in both the processes and products of professional work.

We furthermore maintained that our work was consistent with the best of our abilities, and that we respected any subjects or people involved with our work. This included when we demoed the toy to others, we couldn't leave electrical connections exposed and made sure parts couldn't harm anyone.

ACM 2-6 Perform work only in areas of competence.

Finally, we distributed work in a way that ensured the best odds of success. This meant that if someone was more qualified than the rest of us at something specific, then they would take the lead and guide the rest. This was important for group safety and for the product's safety, but also was more efficient in the long run as they mentored the other groupmates, and we learned faster. Important judgements were only made in competence, as they could lead to failures or setbacks in our project.

CPSC Regulations

The Consumer Product Safety Commission (CPSC) defines guidelines for toys and products in the United States. While these guidelines are for children's toys, we chose to follow some of them for our cat toy; pet toys are not federally regulated for safety[7]. Primarily speaking, we are following the toxicology guidelines, which state we must avoid using unsafe materials in terms of biohazards and bacteria[5]. This applies in our choice of filament for printing our toy shell, PLA, but also for covering our individual parts such as the lithium ion battery. The CPSC also defines electrical requirements which we followed as we only have a maximum voltage of 5V being distributed to our functioning parts[6]. Sound producing toys are addressed in these guidelines as well, wherein we avoided high frequencies of 1000 Hz and above, and volumes that may pose a danger to cats and people[5].

References

[1] PetsRadar, "Best automated cat toys 2024 to keep your cat entertained," [Online]. Available: https://www.petsradar.com/buying-guide/best-automated-cat-toys. [Accessed: 19-Sep-2024].

[2] Pros Versus Cons, "Self-driving toy for your cat: Pros and Cons," [Online]. Available: https://pros-versus-cons.com/pets/pet-products/self-driving-toy-for-your-cat/. [Accessed: 19-Sep-2024].

[3] IEEE, "IEEE Policies - Section 7-8 - IEEE Code of Ethics," [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 19-Sep-2024].

[4] ACM, "ACM Code of Ethics and Professional Conduct," [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 19-Sep-2024].

[5] U.S. Consumer Product Safety Commission, "Toy Safety Business Education," [Online]. Available: https://www.cpsc.gov/Business--Manufacturing/Business-Education/Toy-Safety. [Accessed: 19-Sep-2024].

[6] U.S. Government, "Requirements for Electrically Operated Toys or Other Electrically Operated Articles Intended for Use by Children," Electronic Code of Federal Regulations, [Online]. Available: https://www.ecfr.gov/current/title-16/chapter-II/subchapter-C/part-1505. [Accessed: 19-Sep-2024].

[7] Center for Pet Safety, "CPS Certified Products," [Online]. Available: https://www.centerforpetsafety.org/pet-parents/cps-certified/#:~:text=Pet%20Products%20are%20not%20classified,mandated%20to%20test%20their%20products. [Accessed: 19-Sep-2024].

Appendix

11.1V 2600mAh Li-ion Battery
Datasheet:
https://www.globtek.com/pdf/manual-datasheets/BL2600C1865003S1PGMG.pdf

AZ1117 Voltage Regulator
Datasheet:
https://www.diodes.com/assets/Datasheets/AZ1117I.pdf

SW-420 Vibration Sensor
Information:
https://wiki.seeedstudio.com/Grove-Vibration_Sensor_SW-420/
https://docs.sunfounder.com/projects/ultimate-sensor-kit/en/latest/components_basic/04-component_vibration.html
Datasheet:
https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/Grove_Vibration_Sensor_SW-420_Web.pdf

SG90 Servo Motors
Datasheet:
http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

KY-038 Microphone
Information:
https://sensorkit.joy-it.net/en/sensors/ky-038
Datasheet:
https://kirig.ph/wp-content/uploads/2020/08/KY-038-Joy-IT.pdf

Nema 17 1A Short Body
Information:
https://pages.pbclinear.com/rs/909-BFY-775/images/Data-Sheet-Stepper-Motor-Support.pdf

JST-PH2.5mm Speaker
Information:
https://www.amazon.com/Treedix-Full-Range-Advertising-JST-PH2-5mm-2-Electronic/dp/B0CJNB3CR2/ref=cm_cr_arp_d_product_top?ie=UTF8&th=1

PAM8302 Speaker Amplifier
Datasheet:
https://cdn-shop.adafruit.com/datasheets/PAM8302A.pdf

ATmega328P Microcontroller
Information:
https://www.microchip.com/en-us/product/atmega328p
Datasheet:
https://ww1.microchip.com/downloads/Microcontrollers-ATmega328P_Datasheet.pdf

A4988 Stepper Driver
Information:
https://www.pololu.com/product/1182
Datasheet:
https://www.pololu.com/file/0j450/a4988_dmos_microstepping_driver_with_translator.pdf

VL53L1X Laser Sensor
Information:
https://www.st.com/en/imaging-and-photonics-solutions/vl53l1x.html
Datasheet:
https://www.st.com/resource/en/datasheet/vl53l1x.pdf

CYT1091 Latching Button
Information:
https://www.amazon.com/Cylewet-Self-Locking-Latching-Button-CYT1091/dp/B075VBV4QH