STICKY PIXELS:

AN OFFICE SUPPLY SERENADE!

Interactive Qualifying Project Report completed in partial fulfillment
of the Bachelor of Science degree at
Worcester Polytechnic Institute, Worcester, MA

Submitted to:
Professor Joshua Rosenstock

Seth Crocker ______________________

Christopher Earley ______________________

Dylan James ______________________

Nick Smith ______________________

Submitted May 5, 2010

____
Major Advisor Signature

Department of Humanities & Arts, WPI

Abstract

Our project objective was to create a multimedia art installation that is interactive and adapts/responds to users and/or to its environment. The final installation, Sticky Pixels: An Office Supply Serenade, allows multiple participants to create dynamic electronic music using colored sticky-notes and a specially designed robotic control system. This report comprehensively covers each step of the development and creation of our installation and details our time at the Boston Museum of Science displaying the final product to the general public.

Authorship

Abstract	Chris Earley
Executive Summary	Seth Crocker
	Chris Earley
	Nick Smith
Installation Summary	Chris Earley
Background	Chris Earley
Brainstorming	Chris Earley
Methodology Intro	Seth Crocker
The Wall	Nick Smith
Technical Overview	Seth Crocker
Hektor Clone	Nick Smith
Sweeping robot	Nick Smith
Switching Rig	Nick Smith
Software	Dylan James
Audio Generation	Dylan James
Centralization	Dylan James
Audio Producer	Chris Earley
Driving System	Dylan James
Museum of Science Experience	Seth Crocker
Project Postmortem	Nick Smith
Conclusion	Dylan James
Image Appendix	Seth Crocker
Appendix A	Seth Crocker

Table of Contents

Table of Figures

Executive Summary

Figure 1: The installation in action at the Boston Museum of Science

Over the past twenty years the field of interactive digital art has spread from relative obscurity to the central focus of numerous worldwide exhibitions[1] and galleries[2]. Unlike its renaissance counterparts, interactive art demands a varied skill set and nuanced execution to appropriately describe the desired concept within the constraints of the digitally-controlled medium. The meaning of any artwork, digital or otherwise, relies on the juxtaposition of the artist's purpose for creating the work and the participant's personal interpretation. This blending of personal ego and artistic fore-sight can make or break the strength of the interactive bond between user(s) and the created system. The process of designing a worthwhile idea and then implementing it with the digital and mechanical trappings it needs to communicate was the

[1] http://www.aec.at/festival_about_en.php
[2] Austin Museum of Digital Art: http://www.amoda.org/

central focus of the Interactive Public Art team as we progressed through the stages of development.

Our final installation, “Sticky Pixels: An Office Supply Serenade” is an interactive system that allows multiple participants to create dynamic music using colored sticky-notes and a specially designed robotic control system. (See figure 1 for an image of the final installation in action) It was created with the hope that during its operation, people of different ages, colors, and creeds would work together in a collaborative environment with the sole goal of creating music in a communal environment.

Wall design and Purpose

The Wall was the most important part of our project; in fact it was almost our entire project. The Wall is the structure to which everything else was attached and the part that people where going to come into contact with. We needed some structure that could stand on its own that people could come up to and put their post-it notes on, so we decided to build a board that was going to closely resemble a self standing whiteboard. We designed a wall that was the size of a standard sheet of plywood, which is 5’ by 8’ and we deigned legs that would allow it to be free standing. We made the board from 2x4’s and the face was plywood with a sheet of finish particle board for the face.

Once the board was designed and finished we then began to furnish it with all the peripherals. The other components we added where a safety light, speakers for sound, laptop holder, power supply and holder, stepper motors. All this was required so that we could have people come up to the board and put post-it notes on it and then in turn we could generate music from it.

Wall Plotting Robot

The core concept of our installation was the generation of music through human interaction. This meant that in some way the activities of individuals in the real world had to be translated into digital information which could be processed by our music making algorithm. When we decided to use post-it notes and their colors as the medium for our project, the obvious choice for this physical to computer interface became a webcam. A webcam is cheap and easy to use, making it perfect for our short development period. The only question that remand was how to attach the webcam to the installation. Mounting the camera rigidly opposite the board such that its entire breadth was encompassed was one possibility. However this presented a number of issues, not the least of which was the difficulty in moving the installation to different locations. Also, this plan lacked the level of user involvement we were striving for. We wanted the audience to be able to see directly how their contributions added to the compilation as a whole, which we did not believe would be achievable without a more tangible system. The decision was made, therefore, to mount the camera on an actuated platform and have it travel across the board, much like the read-head of a record player, sampling color data of post-it notes as it went. The challenge became developing a system that would allow fluid motion across 4' by 8' of space. Taking inspiration from preexisting art installations, we developed a system wherein the camera was suspended by timing belt that was reeled in or doled out by stepper motors on the top of the board on opposite ends. By coordinating the motions of these two motors, we could achieve the rectilinear motion required of the camera.

Sweeper Robot

In order to make our installation truly dynamic, as well as to fulfill the integral artistic concepts of destruction and renewal, we needed a system that would be able to remove the creations of the audience and make the board ready for new contributions. Automation of this system would allow for the project to be self sufficient, as well as instill the completed project with some of the mystification every good technological art installation should have. However, the development of this system was no small task, as the very freedom we encouraged our audience to express demanded a robust and encompassing solution. After some deliberation, we determined the best course of action to be a solid metal beam with an attached plastic blade—not so dissimilar from a car's windshield wiper—that swept across the length of the board, removing post-it notes as it went. We actuated the system with a wheeled bracket on the top of the board, driven by a 24 volt geared motor. The system was controlled electrically via the same system purchased to handle the stepper motors. One of the digital outputs off of this board was connected to a semiconductor which controlled the output of an 110V AC to 24V DC converter. Because of the know variability in the resistance the sweeper system would encounter, and therefore the varying speed of the system between runs, some sort of feedback would be required. This was achieved simply by polling the control board's digital input, which was connected to a simple denounce circuit and switch. This switch was connected to one side of the board, which stopped the system near the side and left the main space of the board free for the audience.

Software

We agreed to work on this project in a modular fashion, keeping individual pieces separate so they could function independently. With this in mind, the software running this installation was divided into two separate pieces so that each could function on their own. The two main systems of the installation that needed to be software driven were the audio generation and the driving of the webcam and stepper motors.

The music generation section of this system was critical to creating an enjoyable interactive experience for the users. From the start, we intended on creating multiple different sets of music generation to both keep users interest and to convey different themes. We also were going through iterative development and wanted to be able to constantly be updating, changing, and adding new soundscapes. Because of all these reasons, we chose to keep the generation software as general and flexible as possible. For this, we chose to divide up the sound system into two individual parts – A system which interprets visual data and produces note values, and a system which takes in the note values and produces audio.

For the role of capturing video and interpreting the data, we ended up deciding to use premade libraries from the Processing[3] art software system. Processing is mostly built for image manipulation and as such was a strong choice for the interpretation we would be doing. There are also many libraries around that can be plugged into Processing to allow capturing images from a webcam, so we were able to use it for both of these jobs.

There was no simple or straightforward way of manipulating the webcam image color values into musical notes, especially ones which sound pleasant and resembling music as opposed to just a swarm of noises. We realized this meant that we would probably be constantly changing and updating the way we interpreted the music. On top of that, we also were hoping to allow for multiple themes and interpretations. To do this, we built a set of java classes which would aid in the creation of specific implementations. We started by laying out a system which consisted four layers of objects. Each object would be plugged into other objects, or would plug into another object, or both; this allowed for a high level of versatility in how we chose to interoperate the image data.

The last audio software implementation maped different red-green-blue colors from the captured webcam image to the digital synth instruments and used the varying color intensities of the moving camera to drive the instruments' pitch outputs. From this, the end user could see that

[3] http://processing.org/

different color post-it notes caused different instruments to make sounds and that the more there was of a single color in an area, the more pronounced the corresponding instrument became. Additionally, to keep notes from being sent when the camera passed over the black, or empty sections, of the board a simple highpass filter was used to remove any values lower than a preset luminance threshold stopping any low notes from being passed to the instruments.

For an Audio Producer, we used Jeskola Buzz[4], for creating our synthesized sounds. It is free software that allows plugging in new synthesizers and voices into long patch-chains for simple creation of complex sounds. We had a lot of control over all the voices and it allowed us to link different attributes of the synthesizers to note channels as well, if we saw the need.

The approach we took to the Audio Generation worked out very well for us. It allowed our system to remain modular as we had originally intended. We were able to produce it in pieces and made sure they worked on their own. We were also able to implement different pieces in parallel and in different stages, knowing that we could bring it all together afterwards since we had a predefined interface already laid out.

Driving System

In order to keep the visual data changing, we needed some sort of system to move the webcam across the board. For hardware, we had already decided to use StepperBee[5] control boards to drive two stepper motors with a notched belt running between them. The webcam would be mounted to this belt, so we needed to implement a system to drive these motors to allow us to read all the data off of the board.

From a software standpoint, a lot of the functionality revolved around the StepperBee itself, so the lines between roles weren't as distinct. The system was set up into a few logical tasks, which were mostly independent of each other, but all still interacted with the StepperBee in general and needed some sort of authority and management. Broken down, these tasks were managing StepperBee interactions, handling where to go, and an overall management system.

The largenst objective was to allow us to reference positions on the board as Cartesian points. We wrote functions to move the camera to a given point, which would convert these points into a set number of motor steps using a simple algorithm based off of the Pythagorean

[4] http://www.buzzmachines.com/
[5] http://www.pc-control.co.uk/stepperbee_plus_info.htm

Theorem. This allowed us to drive the camera in a fashion that was easy to think about and simple to comprehend.

We had two different systems for determining where to move the camera, but both ended up writing points to the StepperBee class we made. The first way we moved the camera was selected a random predefined pattern and the second way was user control using a joystick that the user would be given to drive the camera over the board surface.

These two systems would both send their desired targets to the StepperBee in the same way, allowing us to keep the system modular. They were both interfacing with the same code, so if we ended up fixing or updating something in one, it would affect both.

We managed to succeed in keeping our system modular here as well. We allowed for changes and updates to be made to the system. This system was mostly divided into three pieces, each of which only depending on small portions of the ones around it. As our project shifted throughout the course of the year, the system was able to shift and change as well.

Boston Museum of Science Experience

On March 16th an open call for robotic musical installations or related projects was made on the DorkBot Boston mailing list. After a lengthy correspondence with an exhibit coordinator we were invited to install and display our project to the museum-going public. After two weeks of intensive work our project was displayed at the Boston Museum of Science from April 11 to April 17 2010. It was during the Robot Block Party event at the MOS. We went into all of this really excited because this gave us our golden opportunity to show not only our families what we have spent our whole year working on but to see other people interact with our installation which had been the entire point of our project.

However things at the MOS did not go the way we expected. We ran into some issues with our automation and so we decided to make some changes to the project. So instead we gave that control to the spectators. People came up to the board and not only put a post-it note on the board but now used the game controller we had to control the camera and therefore control the music as well. After we got this all set-up about the first hour and a half we allowed people to come up and start interacting with our installation. In the end people liked and enjoyed our installation, especially the kids. People who put a post-it note on the board were delighted to pass the camera over their post-it note and hear the music that their contribution to the board created. We even encouraged the kids to write their name on their post-it so they would know which one was theirs. This was the most enjoyable part of the entire experience for everyone in our group. This was the moment where we said to ourselves "this is what this whole year was about." It was seeing something that we created for people to interact with actually working, and people being thoroughly amused at what music they could create with just some colored post-it notes on a board. We even saw a few of the same people and their kids come back for another try at our installation so they could play their post-it note.

This experience at the MOS was a great learning experience for our whole group. We were faced with technical problems and we found a way to overcome them as a group, so that our project could be enjoyed by the public. We learned a great deal about our group, we learned that we could throw our heads together under the stress of time and the stress of our families watching and we could come up with a solution. In the end giving the control to the people instead of the computer was even better, because it allowed people to get that direct connection we wanted between what they were doing to the board and what that was in turn doing to the music.

Post Mortem

This project faced a number of technical difficulties, primarily in incorporating the many subsystems of our installation. The primary issue preventing full system autonomy as originally envisioned was the lost of calibration in the webcam actuation system. The stepper control board we decided on allowed for only open loop control of the stepper motors. This meant that, giving the motors a command to turn a certain number of steps, there was no way for our system to determine if the motor had actually moved the correct amount. During testing phases these worked quite well, as there wasn't excessive load on either motor, and the control boards performed reliably. However, by the time we displayed our installation at the Museum of Science, the origin of the system (home position of the camera in the upper left corner of the board) was drifting at a rate sometimes exceeding an inch per minute of continuous operation. Because this issue arose so late in the project, we were not able to fully diagnose the cause of the problem. There is some speculation that subsystems lead to brownout issues in our power supplies, or that incorporating addition systems caused subtle deterioration in the performance of the control board. It's even possible that there as the software continued to expand, certain errors were introduced that could account for the missed commands were saw. While the cause of error is unclear, the solution is obvious—as in any system that can't be modeled perfectly—a feedback loop of some sort must be implemented. Solutions are as simple as a periodic recalibration of the software set point with mechanical switches on the installation, or continuous error correction with rotary encoders. Yet despite the need for periodic correction of this one concern, the installation as a whole performed admirably and succeeded in effectively engaging the audience as intended.

Conclusion

The main goal of our project was to design and implement an installation which served some sort of humanitarian purpose. Early on, we decided that this purpose would be bringing people together to work towards one common goal. We wanted to use technology in a way that would get people to interact in person and away from the impersonal sense that we generally see today with things like the internet. As proved by our Museum of Science experience, we were successful in doing so. Though it wasn't exactly our intended audience, people of different age groups and people who didn't know each other would come to the board and interact with our installation together. We had hoped for more person to person interaction as well, but we did manage to break the ice between certain individuals and we succeeded in getting people to interact with both our installation and each other.

In all respects we managed in meeting the goals that were initially laid out to us from the beginning of the term. Our shortcomings laid in the goals that we set out and hoped to achieve on our own, though these were smaller and not critical to the project's functionality. We succeeded in our initial intentions, and produced an operational installation which managed to bring people together, deeming this project a success.

1 Installation Summary

Figure 2: Sticky Pixels during a early test of the user control code

Sticky Pixels: An Office Supply Serenade is an interactive installation that allows multiple participants to create dynamic music using colored sticky-notes and a specially designed robotic control system. The installation itself consists of a four by eight foot black-painted board that is periodically scanned by a robotic webcam mounted to the board's surface (see figure 2). This webcam looks for the color of sticky-notes attached to the board and takes the color information and passes it to a computer for conversion into real-time music. Periodically, a second scraper robot {not shown in the figure above) will activate and clear the sticky-notes from the board so new users have empty space to fill with notes. This process of human addition, musical conversion, and robotic subtraction continues in a cyclical pattern for the operating time of the installation.

1.1 Background

From the very beginning of the project, we knew that we wanted an installation with a purpose, a general concept that we hoped would positively influence the participants over the course of their time with the system. This unifying theme was a source of much discourse during the early phases of the project, mostly due to concerns about the ability to implement an oblique artist statements into a fully realized, functional installation or vice versa. Due to the large volume of topics and mechanical ideas created during this phase, we will not attempt to cover them in detail but instead focus on the concept that led us to our final design.

When it came to finding a niche to exploit for our project, it was beneficial to focus on the everyday things that surround us. By focusing on the humdrum goings-on in life, we received inspiration from subjects that are relevant to our lives. One subject that we were well acquainted with was the idea of technologically mediated isolation, where an individual is more inclined to focus attention on a gadget rather than another member of the local populace. This displacement of social energy from a local network to a disjointed global network was a concept that many of our generation has encountered over the past few years and with some thought, we created a general idea of how to temporarily dissolve the social boundaries through the use of a shared goal.

Inspired by Luke Fischbeck's "make a baby" performance/installation[6], we sought to create a venue for participants to work together as a loose group on a singular task. Since music was a central theme for this project, the idea of having users work to create a dynamic score fit our needs perfectly. The item that was up in the air was just how to accomplish this; how could we inspire and attract users to work with one another in a created environment? With the central idea created, we called upon the many tools and interactive works that came before us for inspiration.

[6] "MAKE A BABY" (interactive performace, 2005-present): http://www.hawksandsparrows.org/mab/

1.2 Brainstorming:

Unlike most IQPs, this project did not supply us with a problem to solve or framework to implement; given the most obtuse of prompts we set out to first find a problem we would solve through the use of art and engineering. When faced with such a limitless set of possibilities, constraints were gravely needed. Over the course of the first few weeks of the project, we spent many hours communicating in person and over e-mail to try to fill out a set of constructive limits for ourselves.

Our process of constraint creation was based on our numerous experiences with team projects, our personal abilities, and the works we had researched for inspiration. From this wealth of information we created a set of realistic limiting factors to work within as we went forward with the design phase of this project. We decided that our final installation must have a defining theme that it aims to express through its operation and interaction with participants. Alongside the theme, the installation itself must contain some physical components since the majority of the team had extensive experience with designing electrical-mechanical systems and otherwise these abilities would go to waste. The majority of the remaining factors fall into either the utilitarian, (it must be transportable, have a pronounced audio element, and much provide immediate interactivity) or the abstract (It must be a novel mode of interaction, should induce a sense of magic or “wow” in the participant, and store information that persists into later interaction sessions). From these guidelines, we then set out to decide on the concept that we wanted to express.

From the interactive works presented in the previous sections, our individual research and interests began to coalesce around a few central themes that after much discussion became our driving concept. With works such as Luke Fischbeck’s “Make a Baby” and Ziggy Campbell’s Cybraphon[7] the dynamic elements of the installations come from not only the interaction of the installation with the users but also from the interpersonal interaction of the users amongst themselves. It was this inspired joining and mixing of random individuals that caught our imagination and we agreed to focus on creating a catalyst for bringing individuals together using music and interaction as our medium.

[7] Cybraphon (interactive installation, 2009–present): http://cybraphon.com/

With the theme chosen and the needed factors set, our next development phase covered the many sessions of idea generation and subsequent sanity-checking that led up to our final design. Our main method of brainstorming consisted of a few days of individual thinking and recording of ideas with little or no filtering followed by a group presentation of the recorded ideas for cross-pollination and eventual discussion. Due to the large volume of concepts and topics presented during this time, we will not focus on presenting all the unused material created during this phase, but ask for those who are interested to look at which contain all summaries of the ideas discussed. That being said, we will instead focus on the direct evolution of concepts that led to our final design.

With the overarching concept of bringing people together, we began to seek out a design that would bring us our desired interaction and create the collaborative environment that we wanted. Originally the idea was to have a type of robot that could crawl along a surface and scan in visual data from user drawings, and to keep the system from getting overcrowded another robot would be enlisted to periodically clear the interaction surface for more open space. This dynamic setup constantly changed in terms of its constituent parts, scale, and setting. The first iteration consisted of a long loop of paper that users were asked to draw on. As the participants made their marks the decorated sections of paper would get pulled into a section of the installation that would read the applied blotches and turn those patterns into music with a large set of data-crunching algorithms performing the heavy musical lifting. The pitfall of this design was that after a large amount of time passed, the surface of the loop would be more user marks than white paper and after many conversations, a suitable method of erasing the ink without significantly altering the form of the design could not be found. Luckily, during this process of realizing the paper loop concept, another idea was made in passing that pulled heavy inspiration from the Hektor[8] project by Jürg Lehni and the process of manually applying wallpaper to sheetrock.

At its core, the proposed idea kept with the concept of allowing multiple users to doodle on a surface at one moment, but this time, the surface was an expanse of wall monitored by a vertical plotter robot that acquired visual data from the wall using an off-the-shelf webcam and complex image-to-sound algorithms. To allow for continuous user additions another more-

[8] Hektor (robotic spray-painting system, 2003-present): http://www.hektor.ch

mechanical seeming plotter robot would be used to plaster white paper over sections of the wall at random intervals, thus covering a subset of user drawing underneath. Once again, after debating the different elements of this design, certain holes began to appear. Unknowns like the drying time of the adhesive used to affix the new paper to the drawing surface, how to keep the adhesive supply going during long sessions with little human upkeep, how to harvest meaningful data from pen scribbles, even the design of the wallpapering robot itself. Since some of these problems could not be easily solved within the time allotted, concessions had to be made. By compartmentalizing the system, we were able to keep elements that could be easily implemented while removing or adjusting others to limit the complexity of the final creation.

From this design revision rose what would become the final design, the vertical plotter webcam robot and user interaction wall elements persisted while the visually-sparse drawing surface and wall-paper robot were left behind for simpler alternatives. This new iteration opted for users to apply stick notes to a interaction surface to generate visually-diverse data for the webcam robot to record and process into sound. Since sticky notes use a mild adhesive to stick to a surface, the roll of the wall-paper robot was replaced with an automated scraper that would travel across the interaction surface, cleaning off a subset of the affixed notes. With the design having been realized, a series of new steps were taken to incrementally bring this idea to life while allowing for many opportunities for learning and adjustments.

2 Methodology

2.1 Intro

This section will describe all of the technical material that went into this project ranging from the physical construction of the board to the software programming. Sticky Pixels took a lot of work in both of the aforementioned aspects to get it up and running. The first part of this section will talk about the physical build of this project and then the last part will talk about what went into the software and programming to get this project to work.

The physical construct and the programming part of Sticky Pixels split our group of four into two teams of two. Chris Earley and Dylan James handled all of the programming that went into Sticky Pixels from the video capture and the music generation to the automation of the stepper motors and the sweeper bot. Chris Earley handled mostly the music generation software that received the color value inputs from the board and programmed the software that made great music from it. Dylan handled all the other aspects such as the interface and all the programming that went into getting multiple different software packages to talk to each other and pass data back and forth.

For the physical part Seth Crocker and Nick Smith handled all of that design and construction. Seth and Nick designed and built the board from wood. Nick came up with the cad models for the stepper motor cases, sweeper bot and the board model. Nick also built the sweeper bot. Seth handled all of the other physical parts such as painting and finishing the board so that it was presentable. He also did all of the wiring and attaching of all of the mounting brackets for the board. The mounting bracket allowed for the speakers, laptop, light and other peripherals to be secured to the board.

So this section will run through in detail all of the pieces of Sticky Pixels that had to come together to bring this abstract concept that our group started with come into a reality. We started off with an idea to bring people together and have them make music and we took that idea and we built a board and did some coding and we made our idea physical.

2.2 The Wall

2.2.1 Requirements

During the construction of this element of our board, we were still dealing with a very abstract vision of our final project. This ambiguity shaped our choices for material and scale. We knew that we needed an upright space that would be capable of holding the medium eventually selected to be read by the scanning portion of the project. We had some idea of the scale we wished to work with, but little idea what systems would have to be added on as our design consolidated. With this in mind, we attempted to design a system that had the maximum number of available mounting points.

Other considerations included the final resting place and the desired audience. At this period we were still toying with the idea of creating a semi-permanent outdoors installation. As such, we needed to consider issues of long term wear and vandalism. Early in the project we also wanted to make sure that we didn't exceed our budget, so cost was an object. Most importantly we needed to make sure that the system wasn't dangerous to the public.

2.2.2 Materials

- 1 8' x 4'-7/16" Sheet of plywood
- 8 12' 2x4 (pine)
- 96" 2x4 (pine)
- ¼ hex nut
- lockwasher
- ¼ x3 ½ hex bolts
- 8 fender washers
- Various wood screws

2.2.3 Tools

- Drill and various bits
- Wood saw
- Level and angles

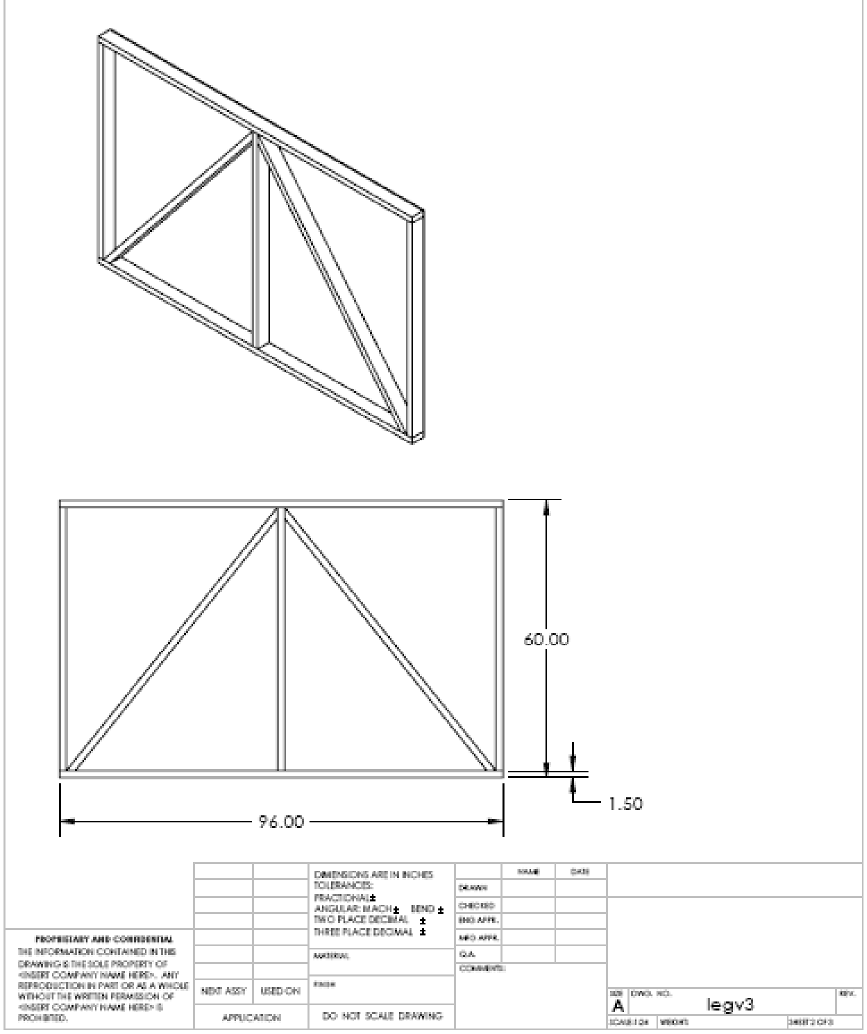
2.2.4 Process

The desire to 'build big' and the necessity to add components on the fly made wood the ideal build material. It was cheap and machinable. The need for an upright smooth surface of large scale made plywood or particle board necessary. Issues of stability and durability made the reinforcement of the board paramount.

Figure 3: Reinforcement of the Board Section

ITEM NO.	QTY.	ANGLE1	ANGLE2	LENGTH
1	1			
2	2	0.00	0.00	96
3	3	0.00	0.00	57
4	2	38.75	51.25	73.09

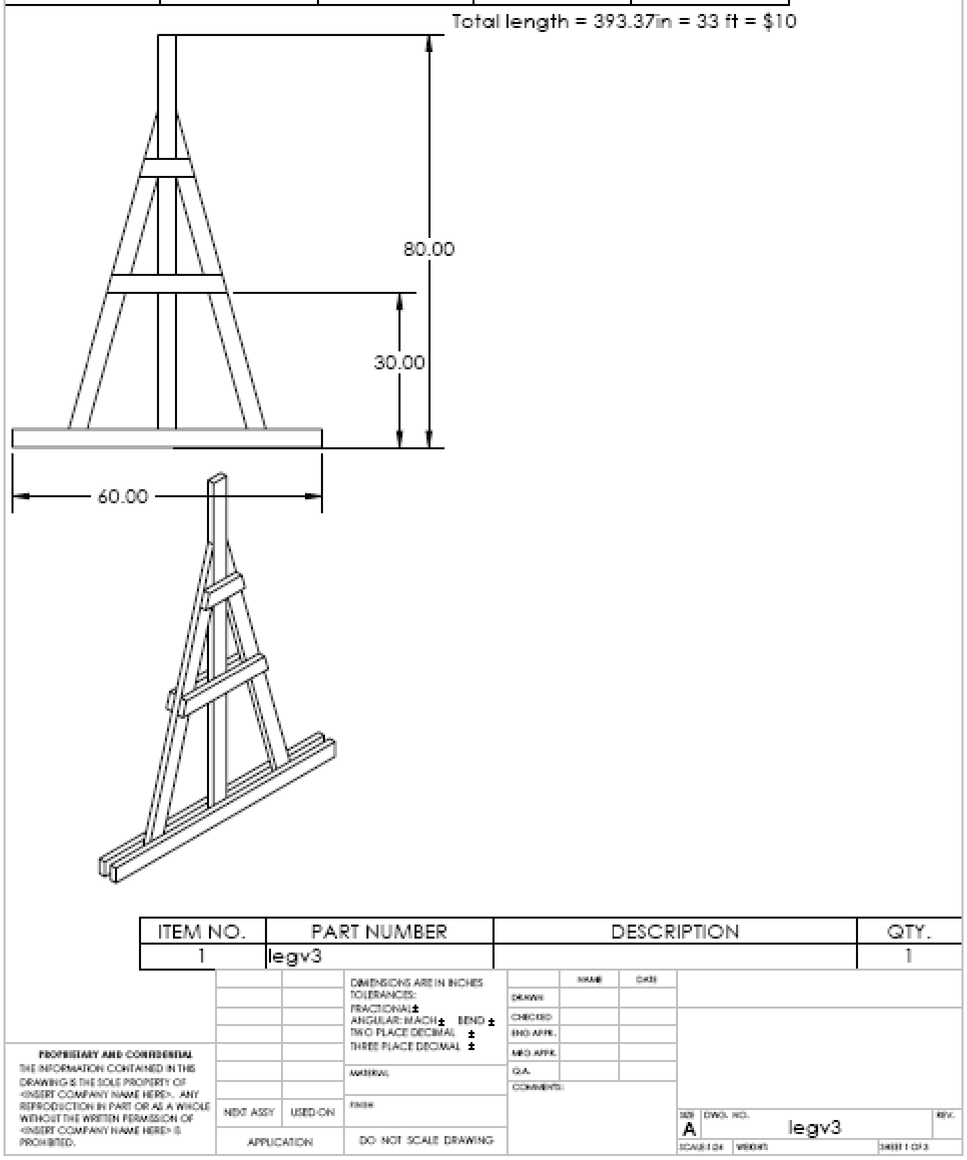
total length = 509.18 inches 42.5 ft = $10 + $22 (plywood) = $35



For simplicities sake we merely framed the 8’ by 4’ plywood board with lengths of 2x4 pine, as pictured in figure 3. This design also had a number of fortuitous aspects. The most significant of which was the large, flat attachment points on the top and bottoms. Secondly was the issue of modularity. From the beginning we knew we would need to move our installation out of our development workshop. The 2x4 on either side of the board provided a surface on which to attach the board legs. With the ability to reduce our installation to 3 distant parts, each weighing less than 50lbs, we felt confident that we would be able to move and assemble our installation with relative easy. Our desired audience was individuals of middle age. As such, we realized that the bulk of the upright working surface should be between 4’ and 8’ feet, around waist and reaching height high on a full grown adult. This meant some sort of riser to move the board up to the desired height and to make sure it was stable enough not to fall on anyone. We decided a simple modified A-frame would suit our purposes. Solidworks calculated the center of mass of our board to be no higher than 45”.

Figure 4: Height and Board Position Considerations

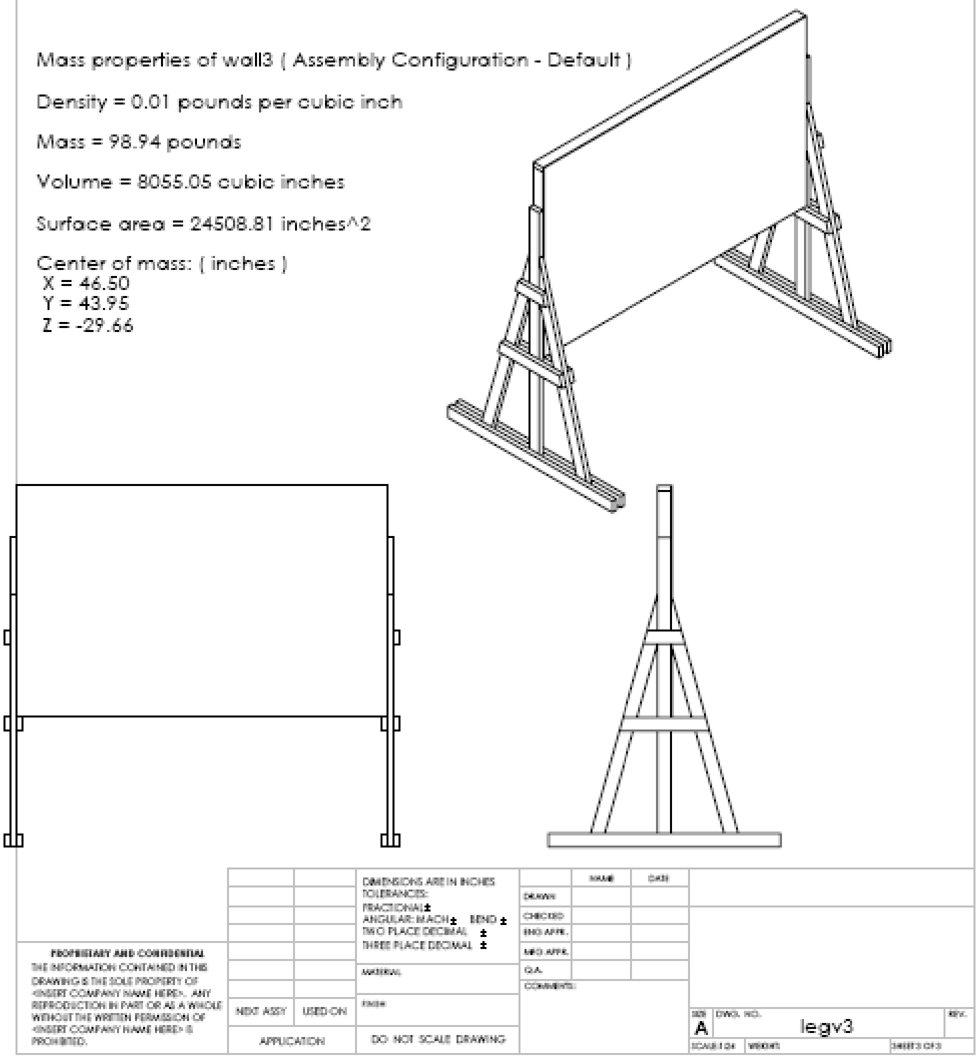
ITEM NO.	ANGLE1	ANGLE2	QTY.	LENGTH
1	0.00	0.00	1	80
2	0.00	0.00	2	60
3	74.33	15.67	2	67.93
4	15.67	15.67	2	23.37
5	15.67	15.67	1	10.77



To ensure that the system was safe, we decided to do a few rudimentary calculations on the system. With a total mass of about 100 lbs and some simple trigonometry and physics, we were able to determine the force needed to knock over our board. The torque generated by the weight of the system is (board weight) * sin(90 – (angle of moment arm and horizontal)) * (moment arm distance) = (100 lb-f) * sin(90 – 56.3) * (54) = 2996 ft-in.

Figure 5: Weight and Cost Calculations

ITEM NO.	PART NUMBER	COST	QTY.
1	wallv2	35	1
2	legv3	$10	2
3	box of wood screws (> 1.5")		1
4	bolts (around 1/4" shold do it) > 3"		>= 4
5	washers and lock washers for bolts		?



If we consider an adult pushing with a force at a height of 5’, the force required to tip the system requires a force of $torque_{gravity}$ / sin(angle of moment arm and horizontal) / (moment $arm_{pushing}$) = 2996 ft-in / 67” / sin(63.4 degrees) = 50 lb-f. We decided this was a reasonable expectation of safety.

These legs were then attached to the main board via the ¼” hex bolts and flange washers, allowing for a reasonable degree of modularity.

2.2.5 Results

The board turned out solid and durable. As time became short towards the end of this project, the decision to use wood proved prudent. Later when we had decided to use post it notes, we were able to treat the board with paint to make a solid surface connection. The only real issue with the board arose later when we began developing the sweeping system. A that point it was discovered that warped lumber from home depot had deformed the plywood surface by almost a inch over 4 feet, which in turn required special design alterations.

2.3 Board Hardware and Electronics

Our project did not just end with the construction of the board. After it was build a bunch of brackets and holders had to be built and attached to the board. It first had to be painted. We decided to go with black because the low luminance values from the color black could easily be ignored by the computer in the music synthesis since any other color would be much brighter than the dark background. The board was first primed and then painted black using latex base paint found at any hardware store. The following peripherals had to be added to the board to support all our hardware:

- The laptop stand
- Speakers
- The sub woofer
- The left and right speakers on either side of the board
- Speaker extension cord
- Led warning light
- Power supply
- Stepper motor wiring harness
- Power cut-off switch housing/ stepper control board holder

The laptop stand was constructed of a 1.5 square foot piece of plywood which was secured to the left side of the board (facing front) by a hinge. Then to keep the board level a rope was tied to the board to support the hinge. This could be untied and then the laptop stand could be folded up for transportation. See figure 6 of a rear view and figure 7 for a side view.

Figure 6: Rear left side of the board

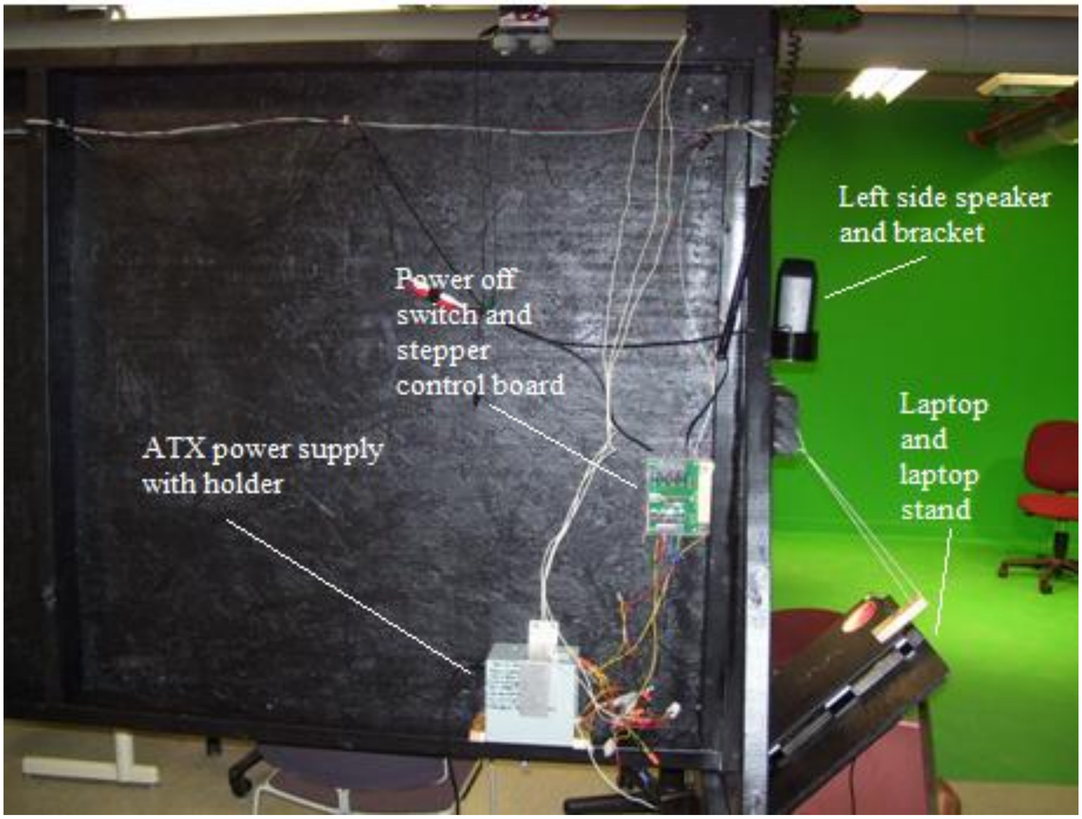
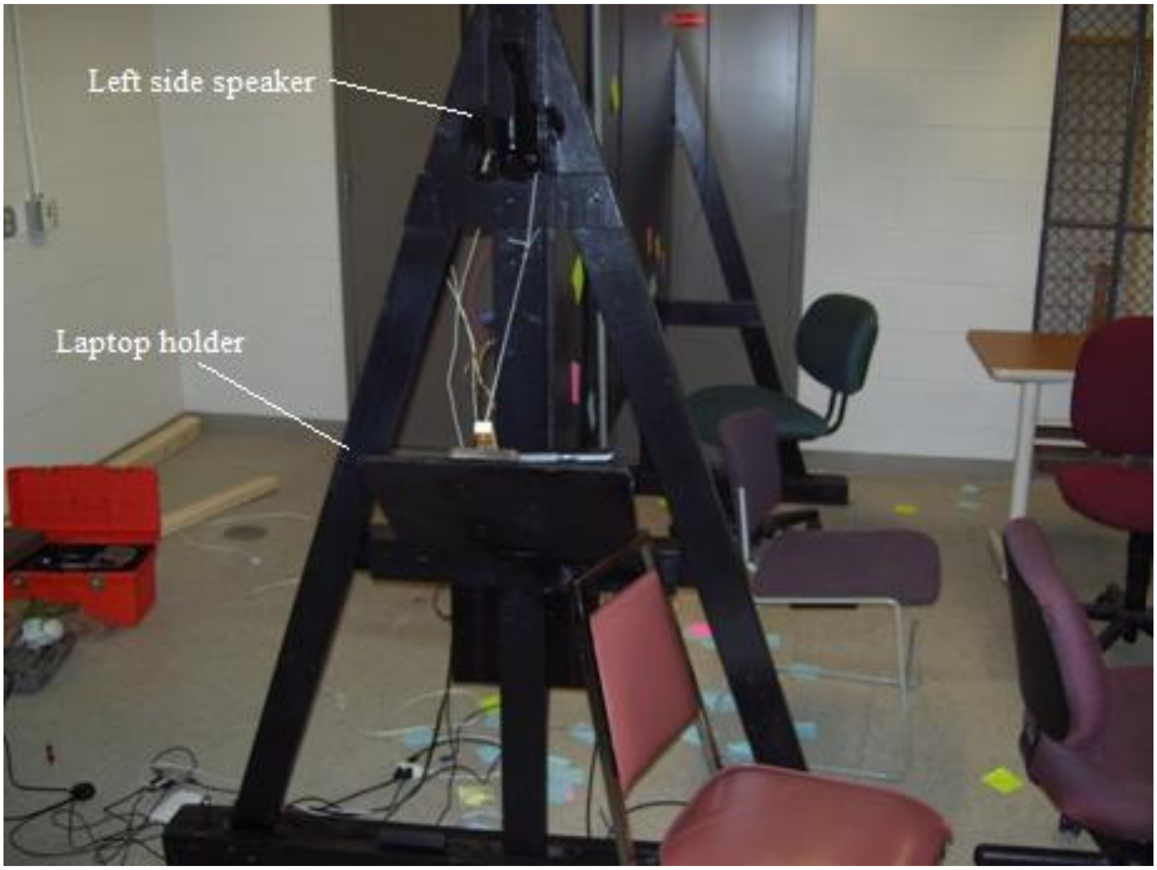
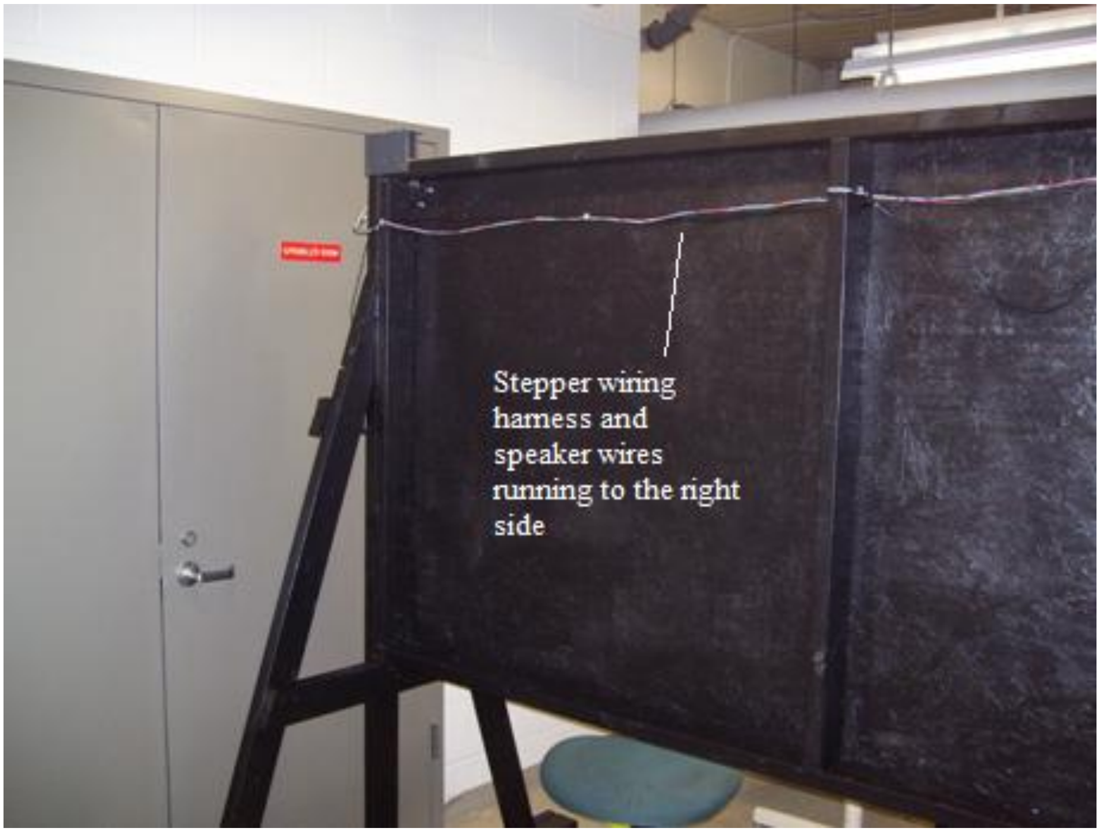


Figure 7: left side of the board



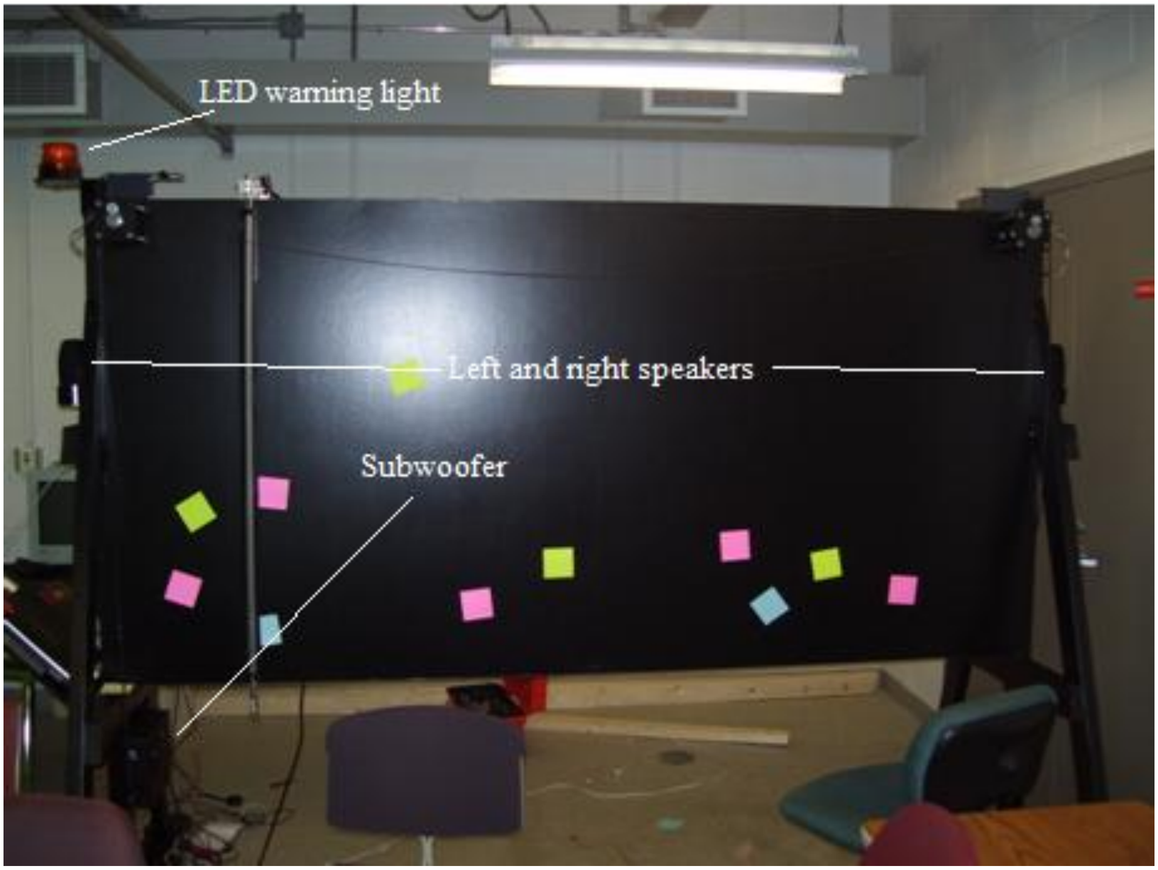
The two small speakers where secured to each side of the board by a 90 degree piece of aluminum which was screwed to a circular piece of ABS plastic that was glued to the inside of a 3" PVC plastic tube. This formed a holder to which the speaker could just be set in and the wires would run to the main subwoofer. The subwoofer was set into a bent aluminum box that was made to conform to the dimensions of the sub. The metal box was then secured to the inside of the left leg (facing front) and the sub was then tied into the box with some electrical tape. The tape was used to allow for easy removal in transportation of the board and also not to damage the subwoofer by drilling holes into it to mount it. The two small speakers can be seen mounted to the board and the subwoofer can be seen in the bottom left hand corner of the board in figure 9

Figure 8: Rear right side of the board



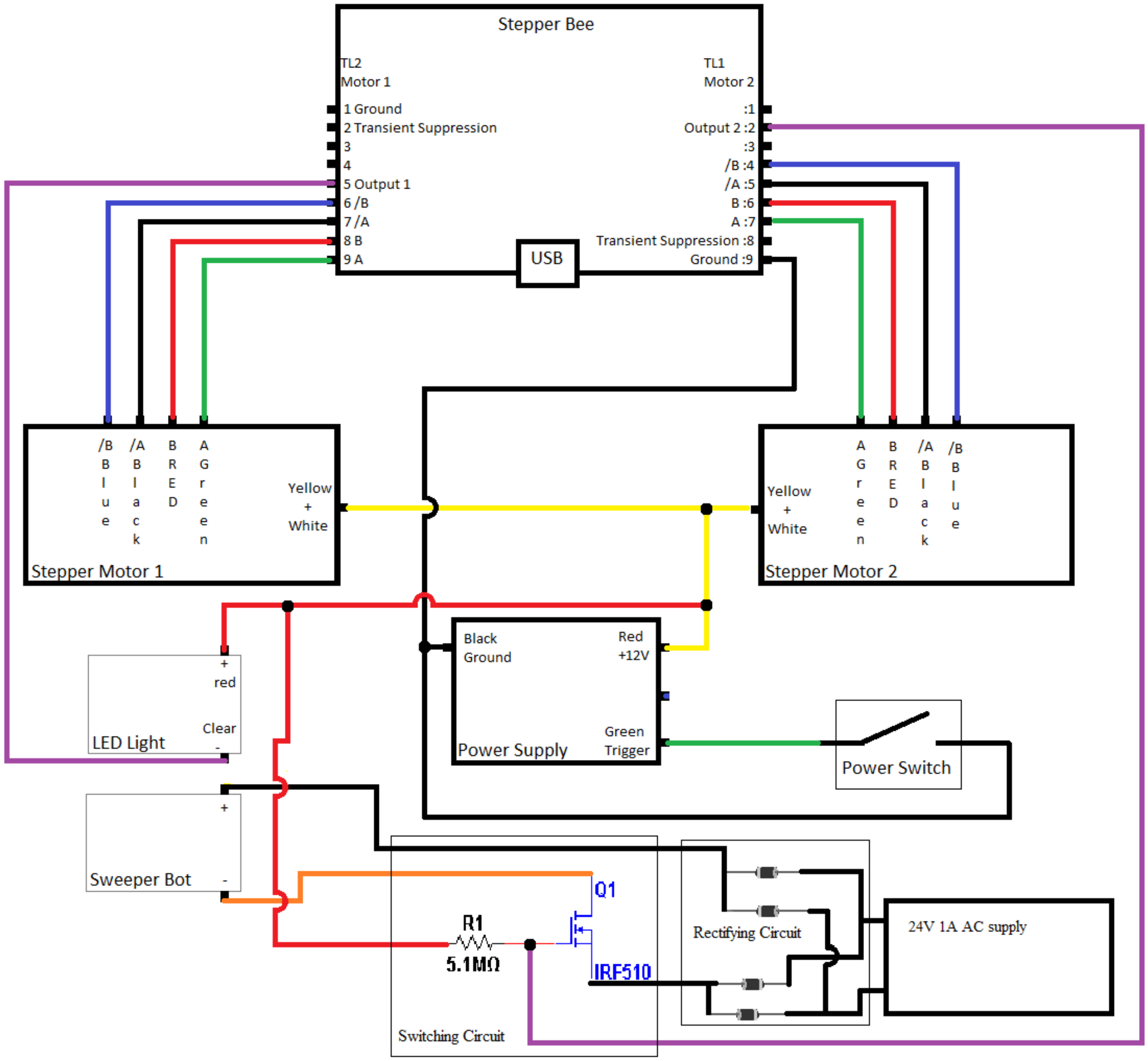
The LED warning light is a 12v strobe light meant for plow trucks and had a 12v cigarette lighter attachment on the end of it and it was removed and wired into the output of our stepper controller board. The bracket that holds it on the board it an L bracket made of aluminum. Then again using screws it was secured to the board. The LED light can be seen in figure 9.

Figure 9: Front of the board



To power out board we used a computer ATX power supply. The 12V leads powered out stepper motors. The scraper bot motor however needed 24V at 0.8A. The ATX power supply we needed only had +12V and -12V power and the -12V was only rated for 0.8A which was not sufficient for two reasons. One is that we would hit that peak current and the power supply would power off and two because the negative -12 created a problem for us when we needed to attached the power MOSFET IRF510 because it was only rated for a 20V difference across the gate (which would be at 0V or +12V) and the source (which would be at -12V). So later we added a 24 AC power supply (not seen in figure 10) which was rated at 1A and had a 0V ground reference. However with that addition a rectifying circuit was needed to change the power supply from AC to DC which is what the motor required. So a full wave rectifying bridge was made from four diodes. To hold all that onto the board again a bent aluminum bracket was made to hold the power supply to the back of the board and two small pieces of wood screwed to the board on either side of it to stop it from sliding left and right. See figure 10 and figure 6 for full representations of the electrical schematic and the resulting physical wiring of the final system.

Figure 10: Wiring Diagram



The last thing that needed to be made was the wiring harness that would connect the steppers to the Stepper Bee control board. The wiring harness was made from 14 gauge wire that had wire connector on the end of it that will allow us to disconnect the motor from the board with ease. The actual stepper board itself was screwed to a acrylic plastic box that was attached to the board that housed the power switch on the inside and the board on the outside. This can be seen in figure 6.

2.4 Hektor Clone

2.4.1 Requirements

The goal of this part of the project was the creation of a system that would be able to move a web camera across our workspace to enable the reading of the board’s content. This meant the navigation of a 4’ by 8’ horizontal space with Cartesian motion. Because we still were discussing methods of board erasing or scrapping, we needed to maintain a certain space between board and read head. And, as previously mentioned we needed to maintain a low price point.

2.4.2 Materials

- ¼” ABS plastic
- 3/8” Aluminum round stock
- ¼”-20 hex bolts
- ¼’-20 nuts
- 4-40 machine screw
- 4-40 machine nut
- Single sided 3/8” trapezoidal neoprene/fiberglass .08” pitch timing belt (MXL)
- ¼ “ ID 1-1/16 ” OD steel bearing
- 1.12 OD Belt pulley
- Soyo 12V 0.68A 125oz-in Unipolar Stepper Motor

2.4.3 Tools

- Machining Mill, various bits

Figure 11: Parts list and Explanation

part name	mcmaster part number	description	amount needed	individual cost	notes	subtotal
plastic sheet	8712K69	ABS plate (1/4" x 4" x 24")	1	8.74		
#4-40 3/4" flat slotted machine screw	90275A113	pack of 100	1	3.49		
4-40 machine screw hex nut	90480A005	pack of 100	1	0.76		
1/4-20 nut	93827A211	pack of 100	1	5.94		
ID 1/4" OD 1" bearing	638K220	per	2	3.88		
1/4"-20 1-1/2" flat machine screw	90273A546	bag of 50	1	7.03		
single sided trapezoidal 3/8" neoprene/fiberglass .08" pitch timing belt (MXL)	7959k22	per foot	20 ft	2.50	cheapest 1/2" width belt is 2.85, but it is urethane (which supposedly is less flexable). strangely enough mcmaster doesn't seem to carry a pulley for this type	

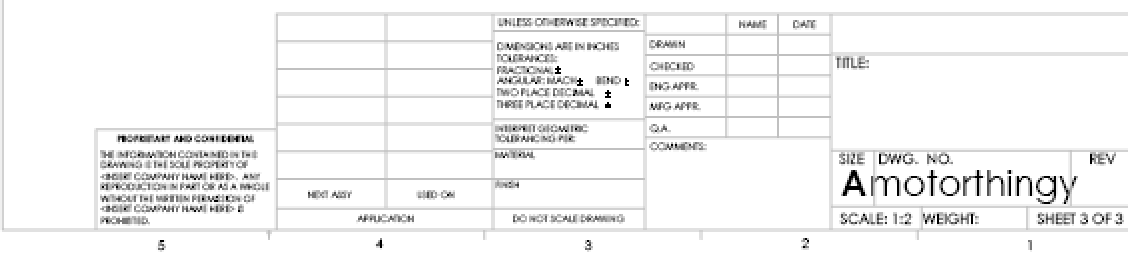
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER: MATERIAL FINISH DO NOT SCALE DRAWING
NEXT ASSY USED ON APPLICATION
NAME DATE DRAWN CHECKED ENG APPR. MFG APPR. Q.A. COMMENTS:
TITLE:
SIZE DWG. NO. REV
A motorthingy
SCALE: 1:2 WEIGHT: SHEET 2 OF 3

PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

5 4 3 2 1

Figure 12: Parts List and Explanation

name	mm part#	description	amount needed	individual cost	notes	subtotal
1/4" .08" MXL series	7959k21	per foot		1.81		
1/4" .2" XL Series	7959k24	per foot		1.82		
MXL .685 OD .08" pulley	1375k39			9.81		
XL 1.12" OD .2" pulley	6495k714			11.66		
1/4" ID, 1-1/16" OD 1/4" width open bearing	6383k25			4.76	open have exposed balls for easy lub. sheilded protect from contaminatnts	
7/8" OD open bearing	6383k14			3.88		



2.4.4 Process

As luck would have it, research revealed a project very similar to what we had in mind. Hektor bot[9] utilizes two stepper motors, timing belt, and the forces of gravity to generate a Cartesian motion for a spray can which is used to generate art. The similarities between our project and the pre-existing hector bot convinced us that we could save time and resources by borrowing components of their design. Moreover, the existence of a functional system that operated on the principles we intended to use would help to ensure we didn't waste any of our time on concepts that were fundamentally unfeasible.

[9] http://www.hektor.ch/

What really attracted us to the Hektor concept however, was its apparent simplicity. Many conventional forms of Cartesian robots would be unfeasible due to budgetary constraints. Even small Cartesian robots are expensive, but when the size and orientation of our desired workspace is taken into account, we knew a less orthodox approach would be necessary.

Stepper motors were chosen as a means of actuation in an attempt to keep system complexity low. It was our hope that by using stepping motors, now feedback system would be necessary and development time would be quicker. The particulars selection of our motor was dictated by worst case camera weight calculations.

Figure 13: Mechanical Dimensions of the Selected Motor

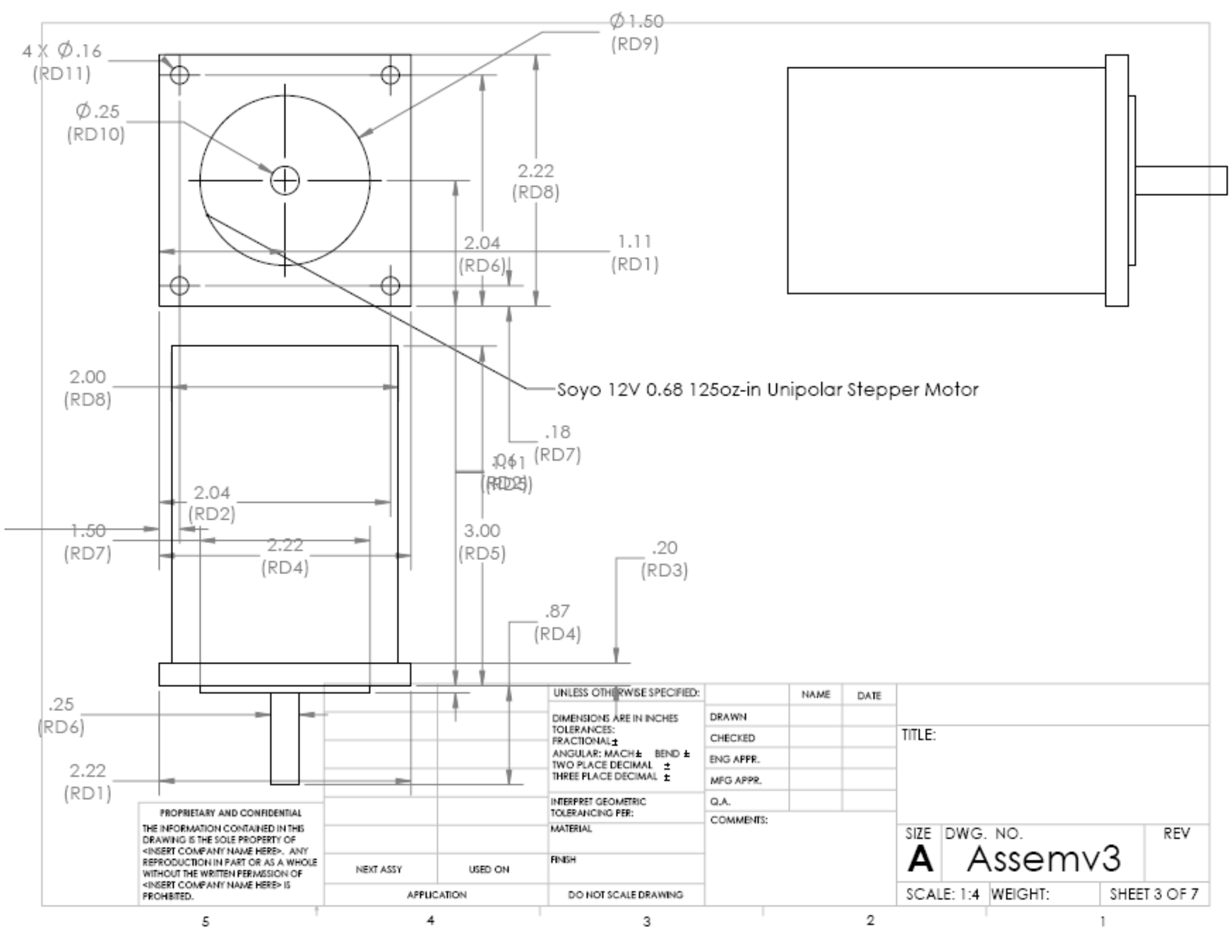


Figure 14: Electrical Characteristics of the Selected Motor

RobotShop INC.

La robotique à votre service

Soyo 12V 0.68A 125oz-in Unipolar Stepper Motor Specifications (double shaft):

Item	Specifications
Step Angle	1.8°
Step Angle Accuracy	±5% (full step, no load)
Resistance Accuracy	±10%
Inductance Accuracy	±20%
Temperature Rise	80°C Max.(rated current,2 phase on)
Ambient Temperature	-20°C~+50°C
Insulation Resistance	100M Ω Min. ,500VDC
Dielectric Strength	500VAC for one minute
Shaft Radial Play	0.02Max. (450 g-load)
Shaft Axial Play	0.08Max. (450 g-load)
Max. radial force	75N (20mm from the Flange)
Max. axial force	10N
Rotation	CW(See from Front Flange)

Model No.	Rated Voltage	Current /Phase	Resistance /Phase	Inductance /Phase	Holding Torque	# of Leads	Rotor Inertia	Weight	Detent Torque
Single Shaft	V	A	Ω	mH	Kg-cm		$g\text{-}cm^2$	kg	g-cm
SY57ST76-0686B	12	0.68	17.7	30	9	6	200	0.72	0.72

Once we had selected our motors, the adaptation of the proprietary Hektor plotting system to our setup was relatively easy. We borrowed the operational principles of the system using the picture and videos posted on the project website but redesigned the mounting system to better suit our limited budget and materials. The principle of operation was simply that a timing belt is sandwiched between actuated pulley and freewheeling bearing in order to keep the belt in contact with the pulley at all times.

Figure 15: Belt and Holding Systems

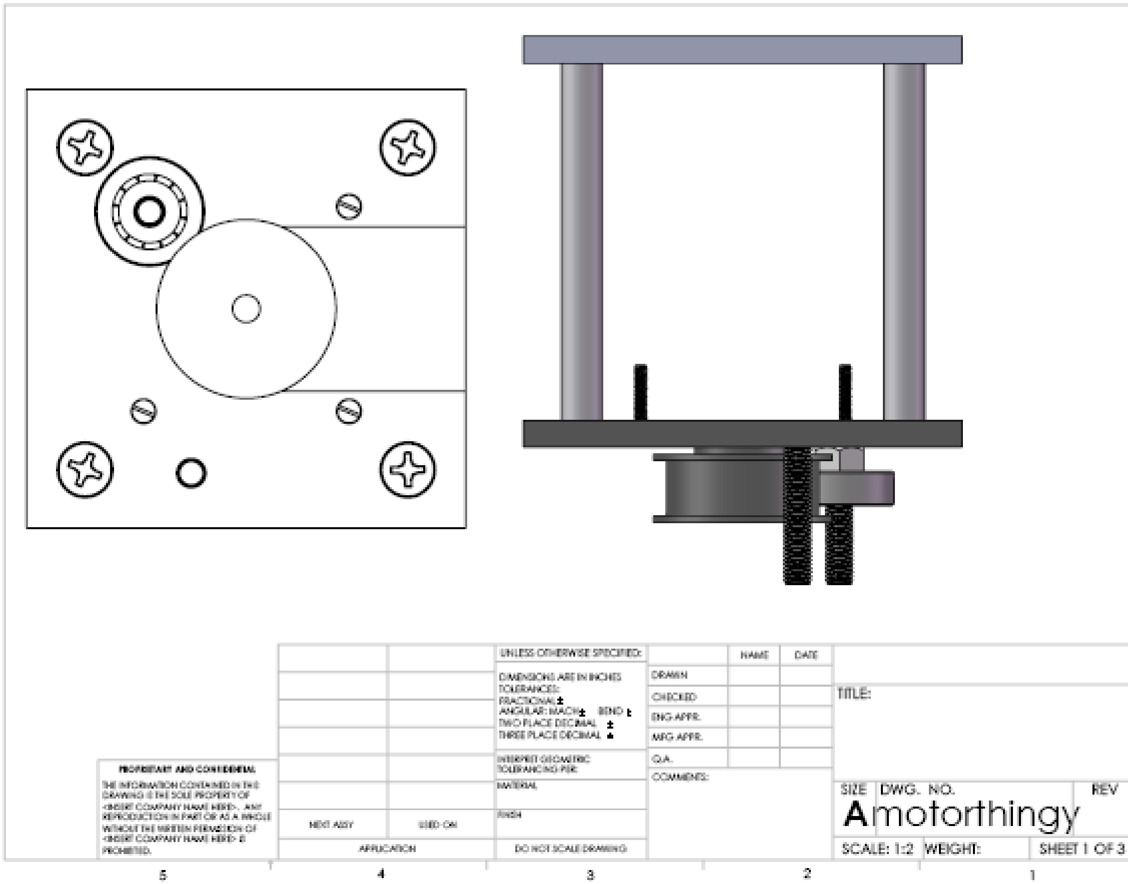


Figure 16: Timing Pulley Belt

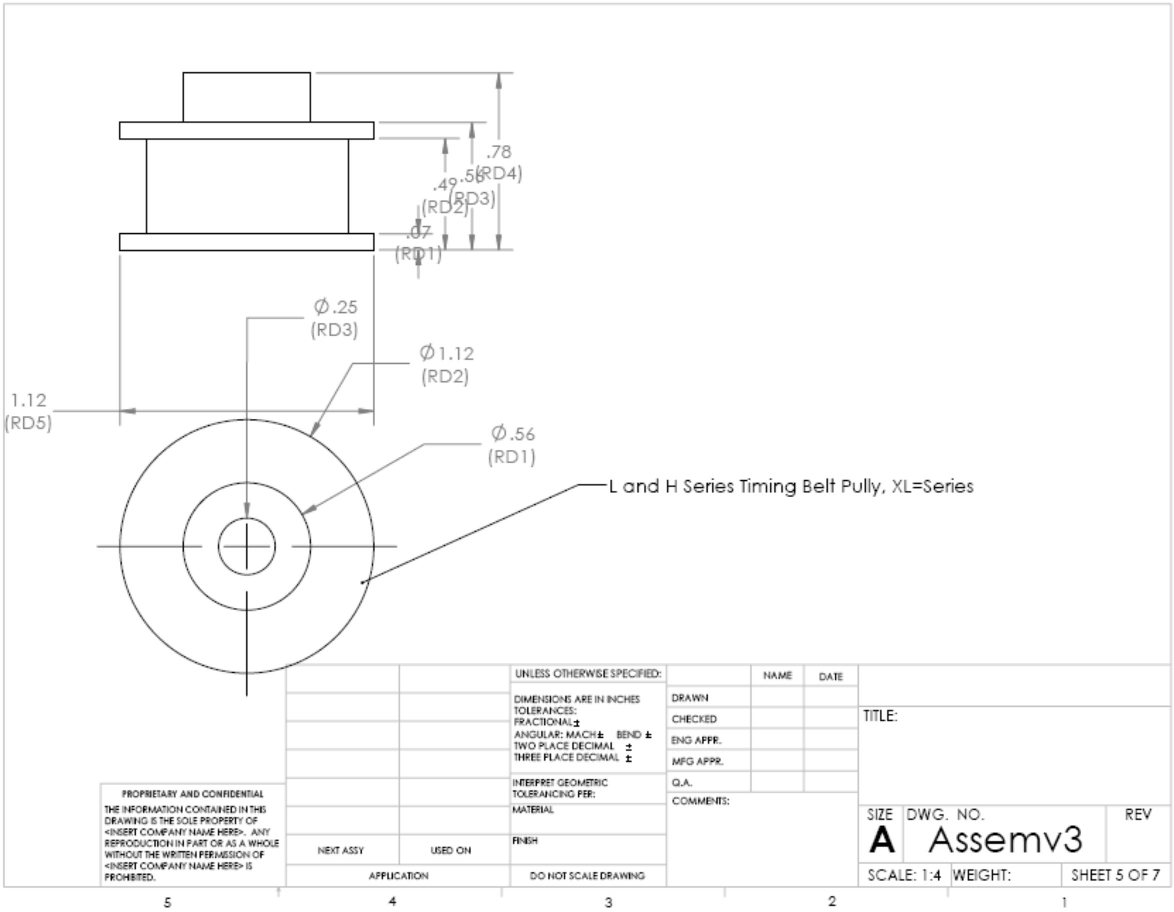
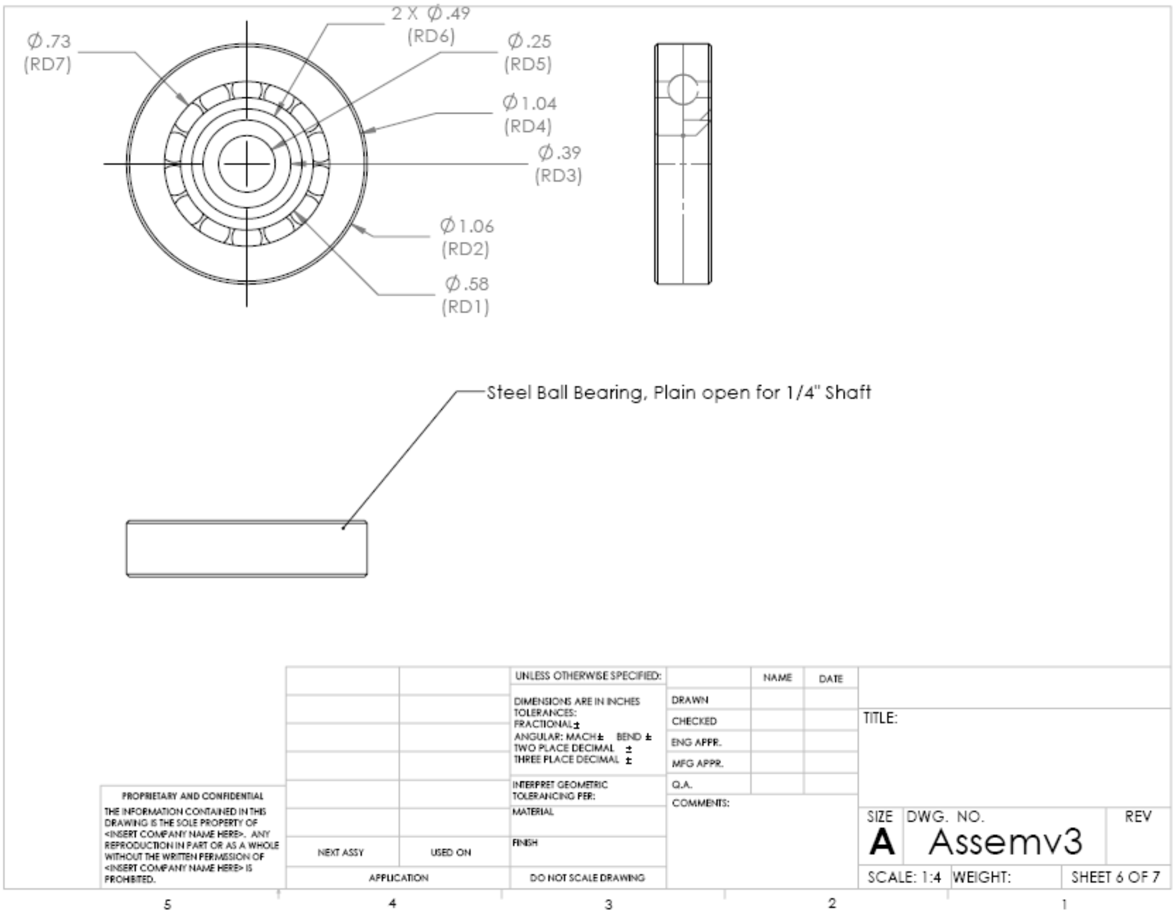
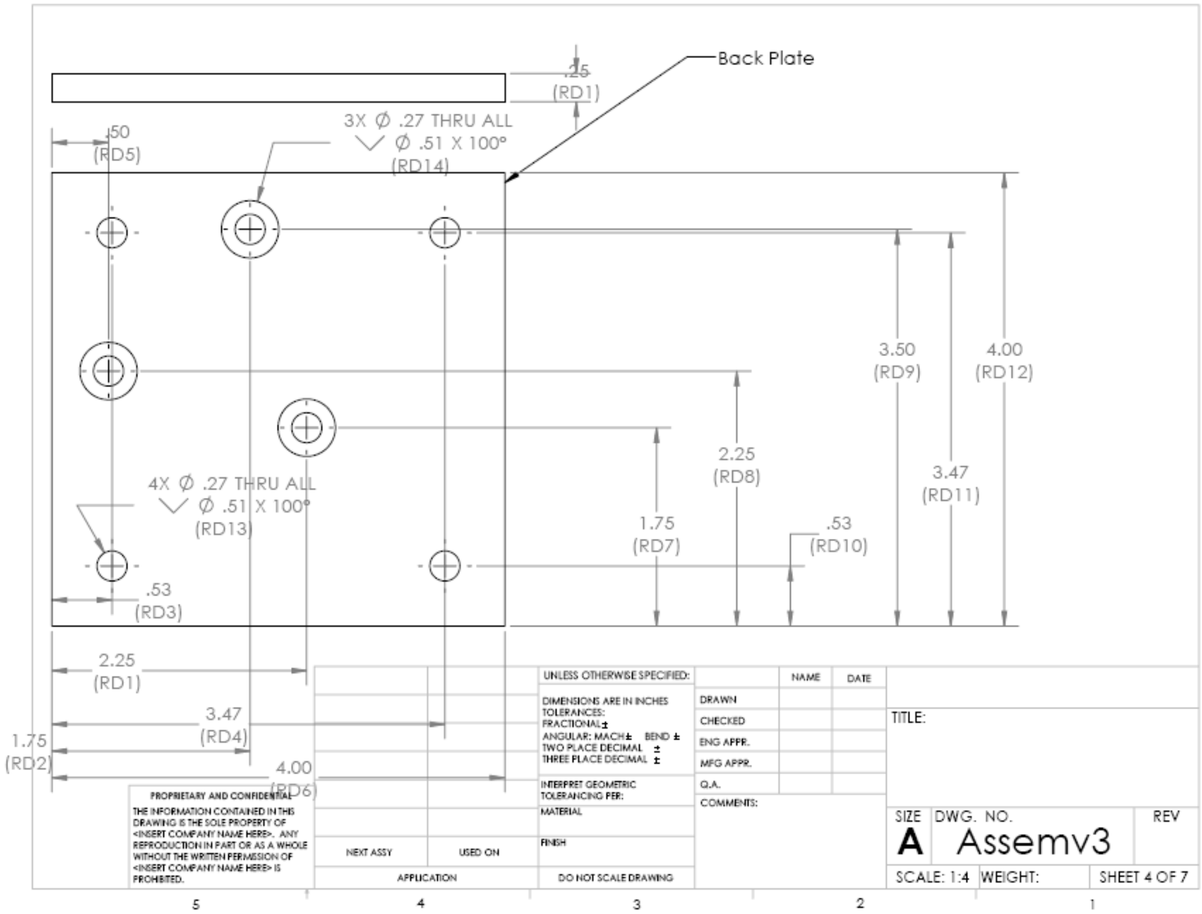


Figure 17: Constraining Bearing



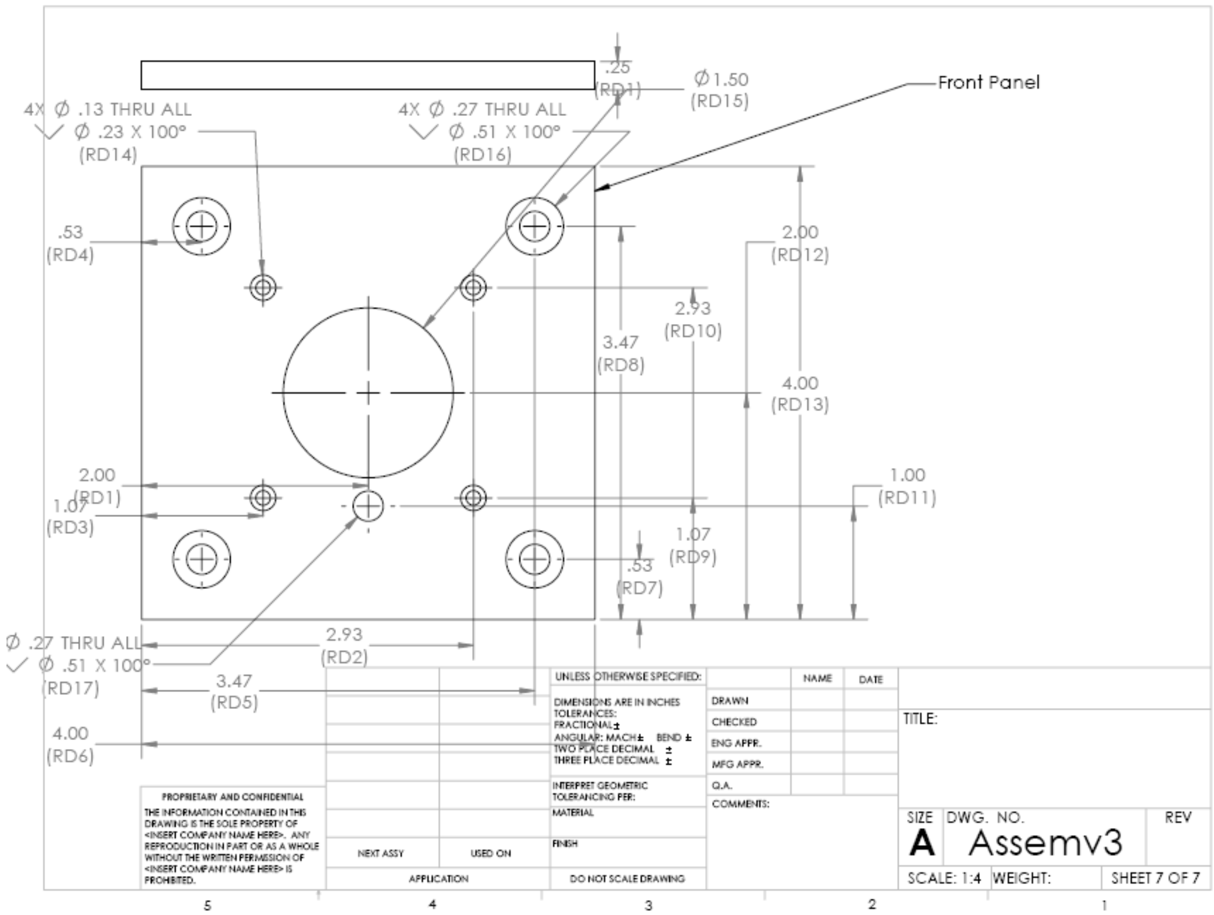
Because of the dimensions of the motor, and because we wanted to make sure that the entire system could be removed from the board, aluminum standoffs were used and a back ABS plate with a specific bolt pattern was created for both modules. The triangular pattern of countersunk holes in figure 18 allowed machine screws to connect this plate to the wood board without interfering with the Motor. The front assembly (consisting of the motor and front plate) could then be connected to the back assembly (the back plate and standoffs already connected to the board).

Figure 18: Back Plate and Bolt Patterns



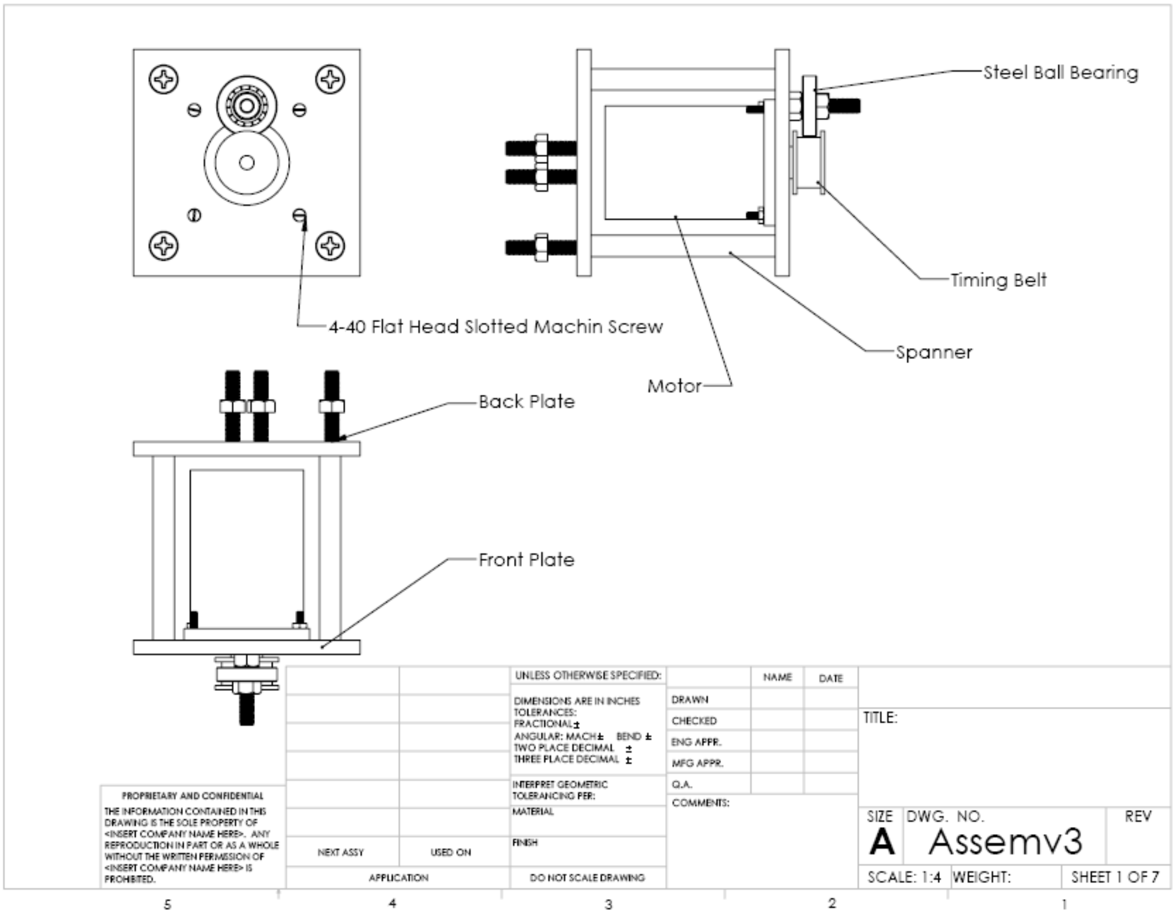
The front face plate, too, was made of ABS. It had the correct bolt patterns for motor, tensioning devices, and standoff drilled into it. The space around the timing belt pulley was left free in case so sort of calibration system was later needed.

Figure 19: Front Panel Design



The system was refined until it was simple enough that it could be machined using only drilling operations, and the entire system was fitted together and attached to the wooden board through the specified bolt pattern.

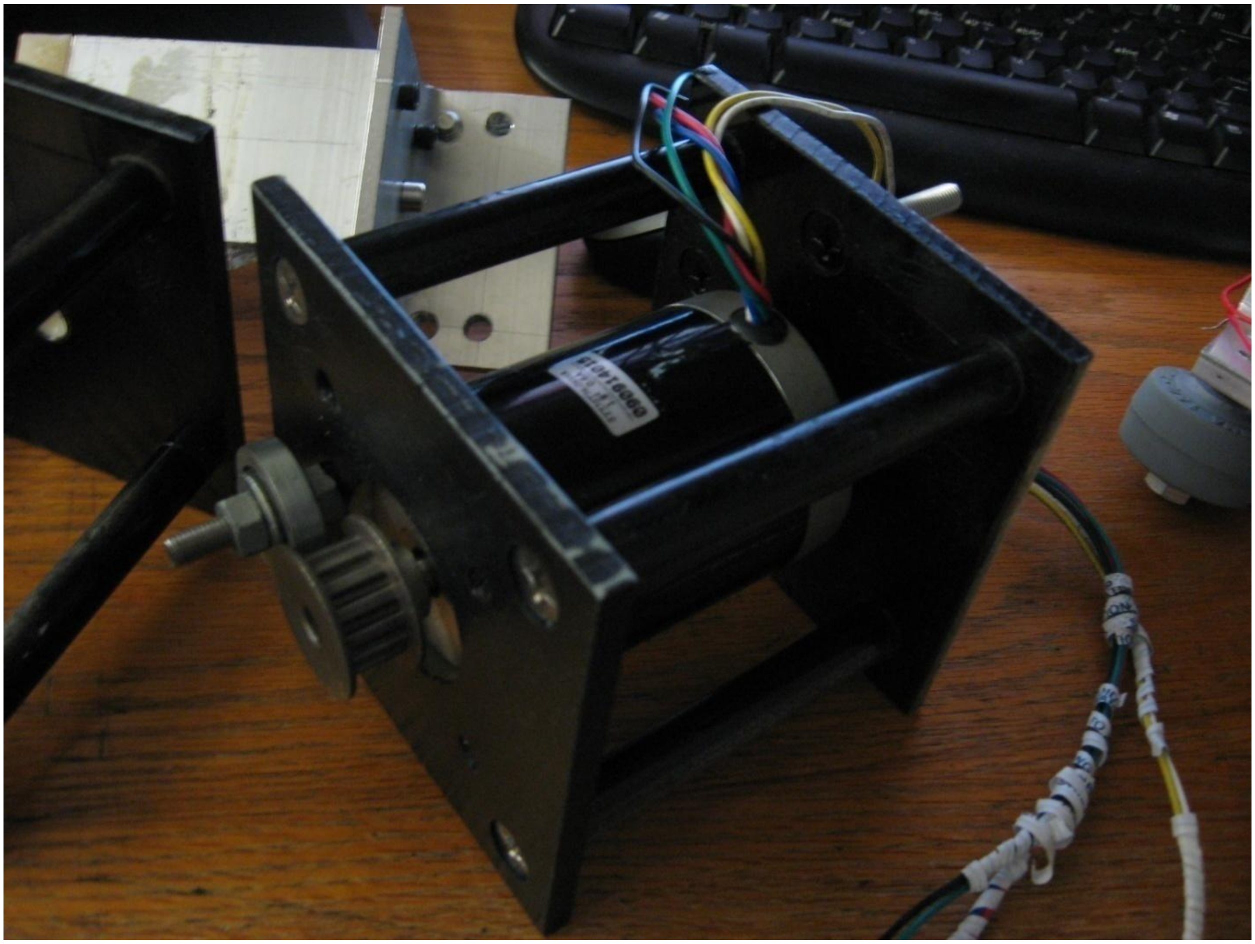
Figure 20: Assembled System



2.4.5 Results

Although countersinking operations in ABS plastic proved to be somewhat problematic during fabrication, the first iteration of design performed well enough that no redesigning was necessary. Belts travelled smoothly and consistently, and the jams or dropped belts we worried about never occurred.

Figure 21: Machined System



2.5 Sweeping robot

2.5.1 Requirements

As it became resolved that we would be using post-it notes on our board, the desire was expressed to have a system that could automatically remove the notes, in order to keep in sight of the original themes of our project.

This meant that we needed a system that could be actuated across 8’ of board while simultaneously removing large quantities of sticky notes. We needed something cheap, durable, and safe.

2.5.2 Materials

- 1/2 “ ABS plastic
- 5/7” rubber wheel
- 5/16”-18 1-3/4” hex bolt
- 5/16”-18 hex nut
- 4-40 machine screw
- 4-40 machine nut
- 3/8” aluminum round stock
- Aluminum u-channel, 3” x 1-3/4”
- architectural aluminum 1/2” x 1/2” 1/8” width angle stock

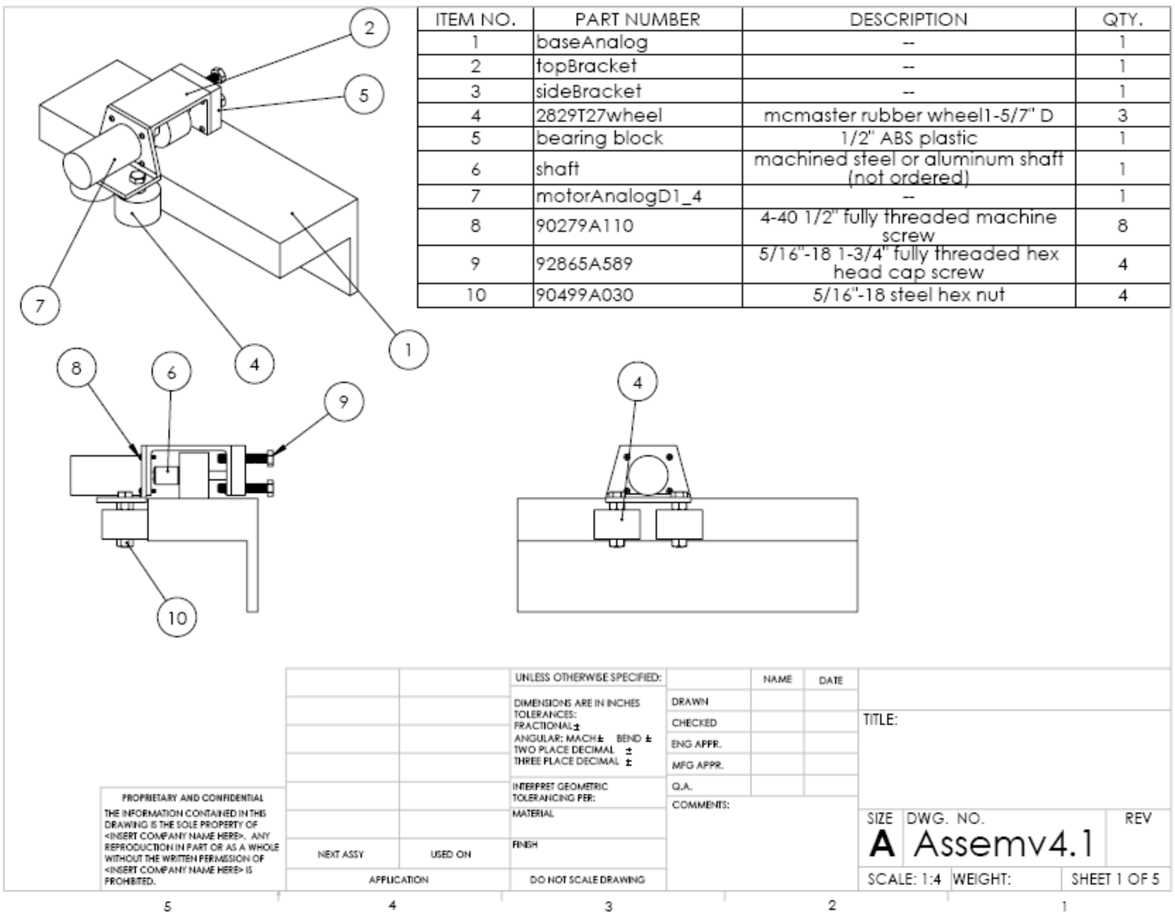
2.5.3 Tools

- machining mill
- machining lathe

2.5.4 Process

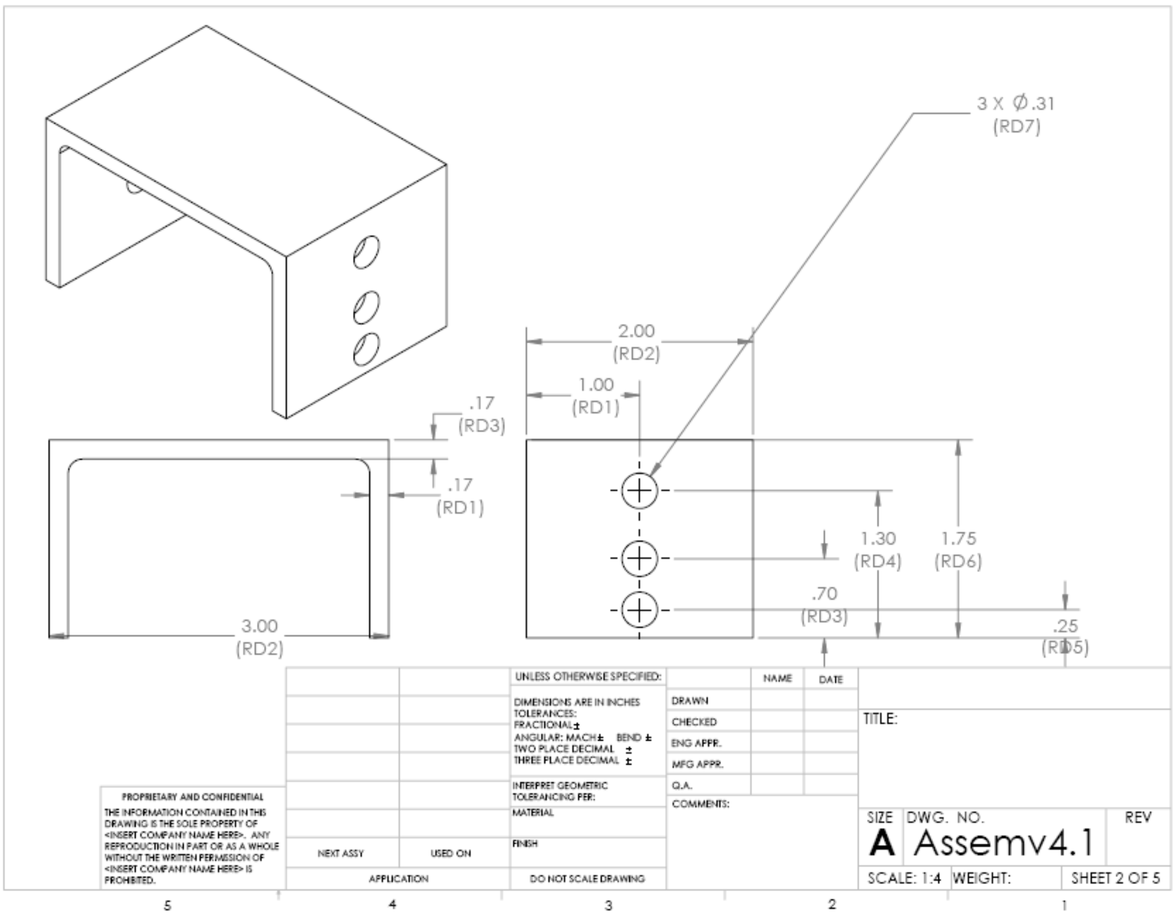
Some testing with post-it notes quickly revealed that the best method for removal was a thin piece of material running across the board at a low angle of attack. Our original plan was to create a rolling bracket to run the lengths of the top and bottom connected by a rigid bar of stock aluminum. The original scraping blade was thin straight plastic blades lining the length of the aluminum L-bar.

Figure 22: Bracket Design



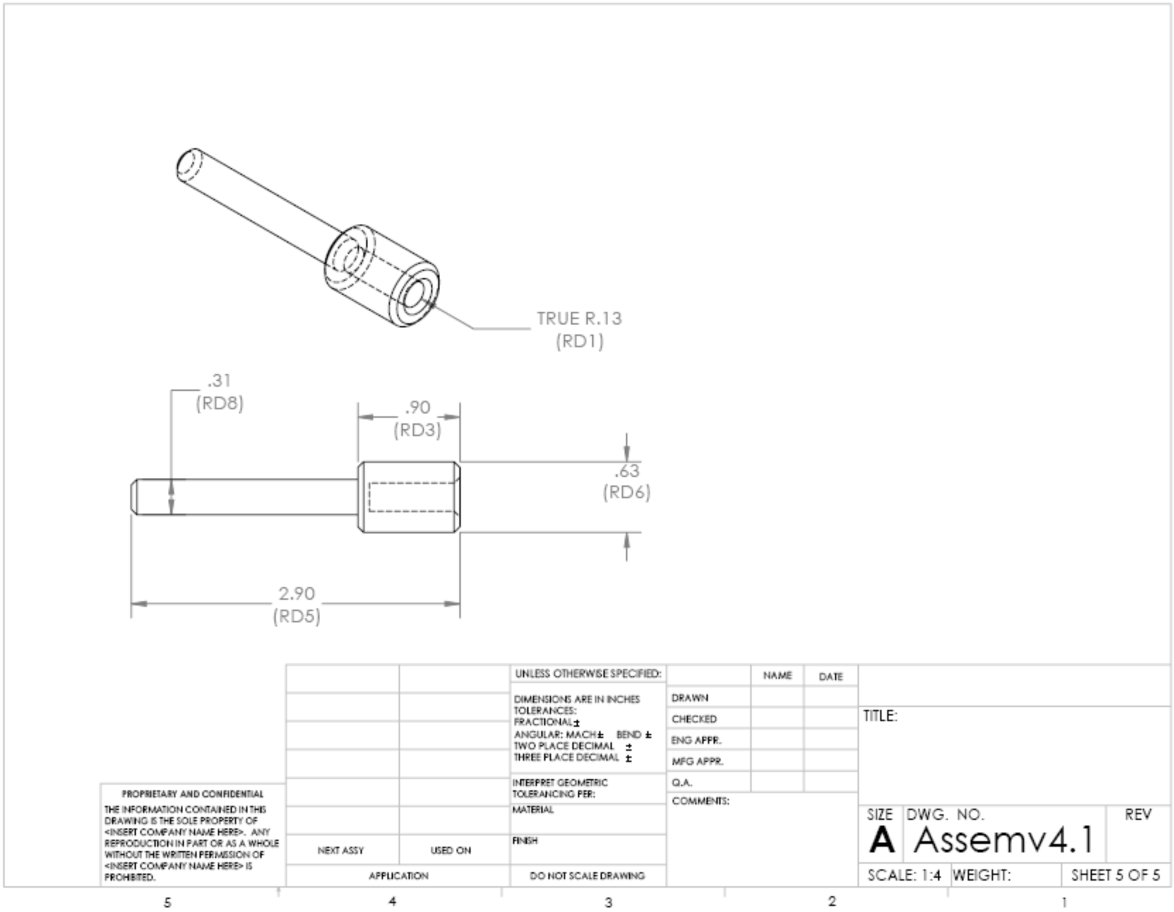
The body of the bracket was created using 3 inch U stock extrusion aluminum. Holes were drilled to allow an actuated shaft for a wheel on the top of the board, as well as for attaching the motor and bearing block.

Figure 23: U channel



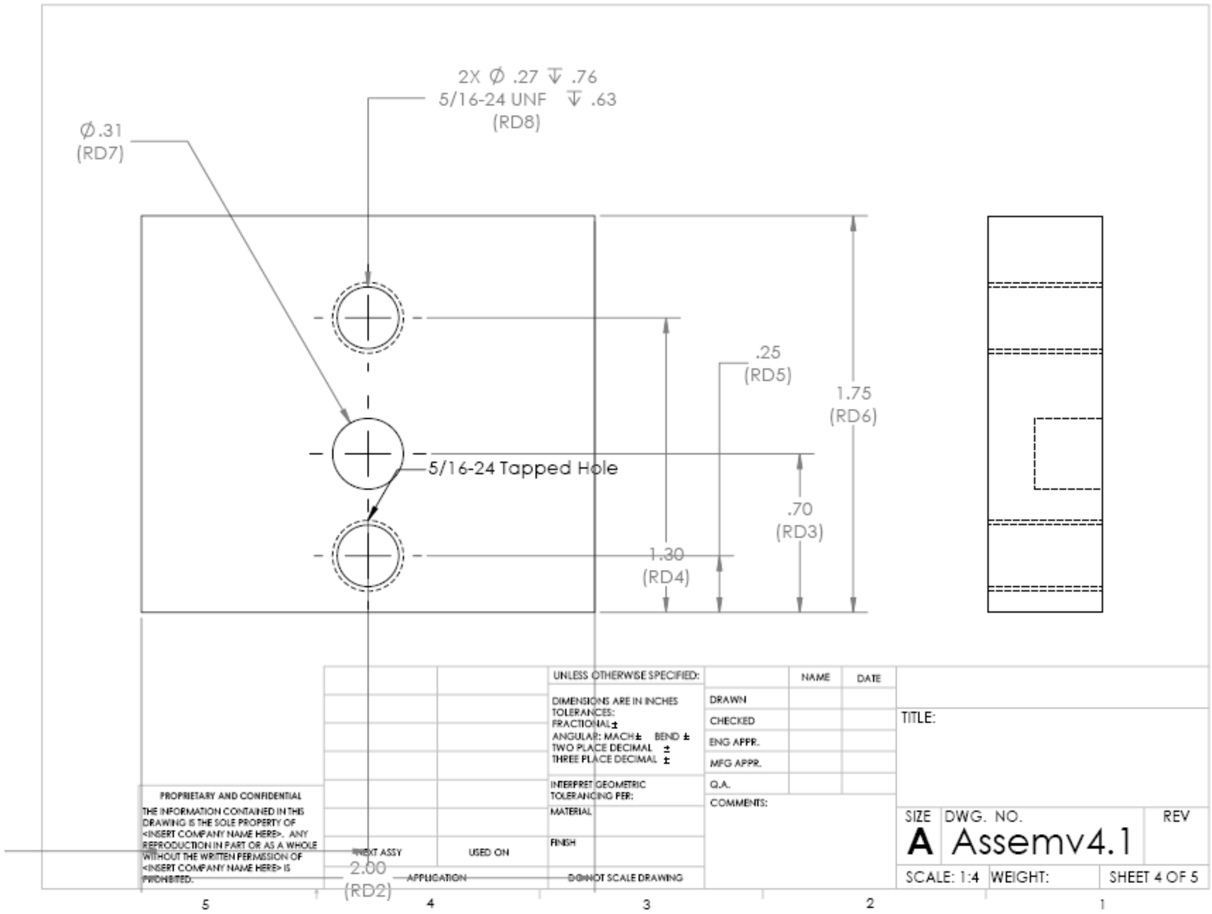
The shaft was stock aluminum rod lathed to convert the motor shaft into the diameter of the rubber wheels. The rubber wheels were affixed to the shaft by glue and a set screw attached the adaptor component to the motor shaft.

Figure 24: Aluminum Shaft



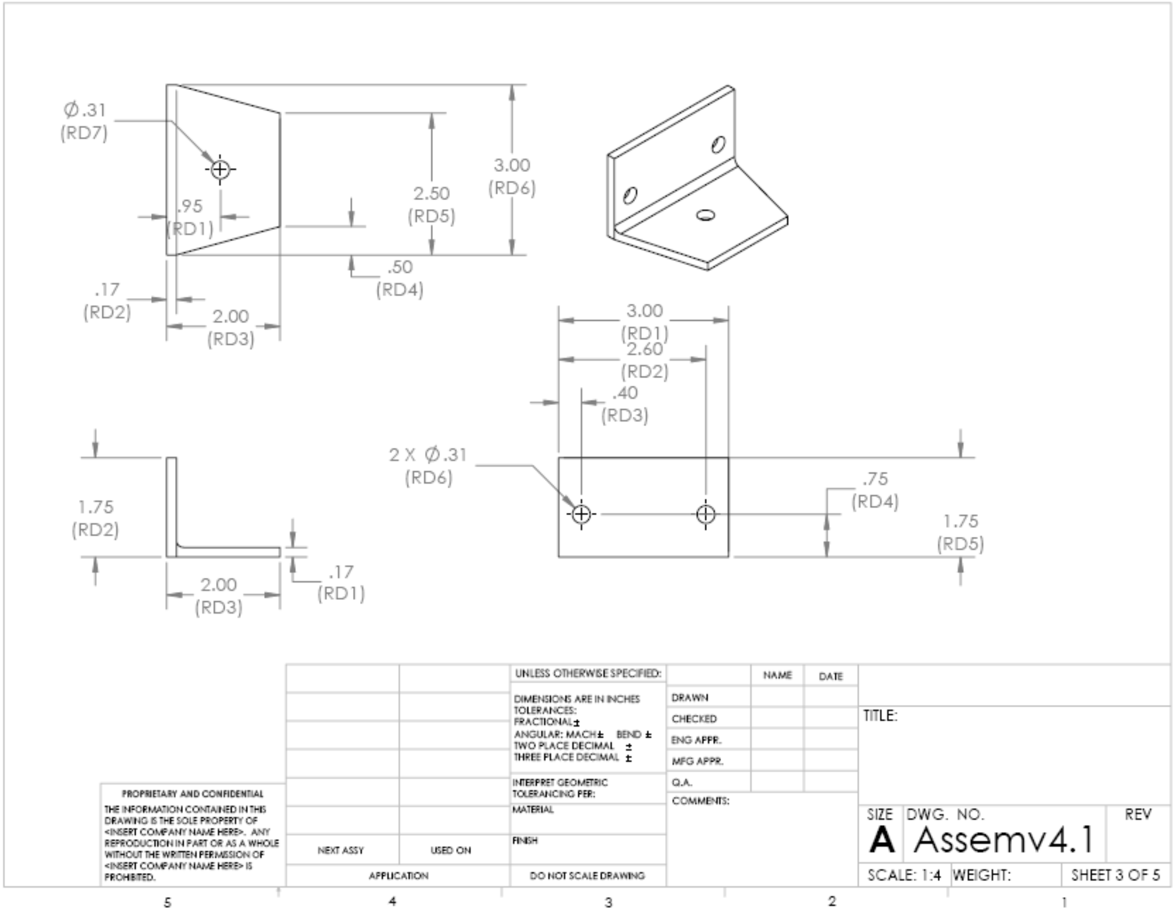
On the side opposite the motor, the shaft rests in a Teflon pillow block. This block also has holes for the attachment of the scrapper.

Figure 25: Teflon Pillow Block



On the side with the motor we used L channel aluminum extrusions with holes in them to hold two axels on which additional rubber wheels rotated freely, providing forwards and backwards constraint.

Figure 26: L channel

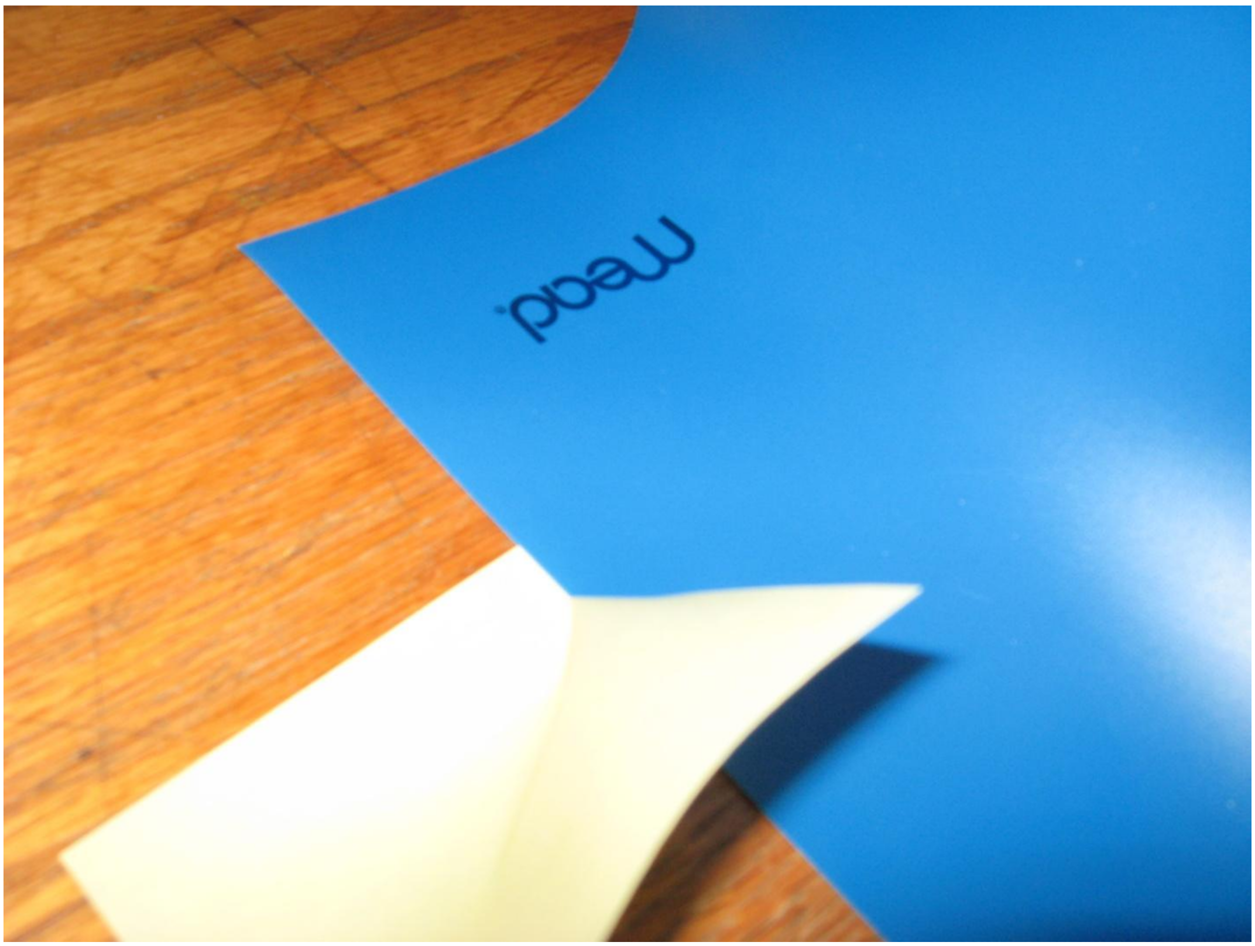


However, difficulties quickly presented themselves. With rollers only on the back of the top bracket, the tensioning of the bottom bracket caused irresolvable frictional issues between the scraping blade and the surface of the board. However, if the bottom bracket was removed the distortions in the plywood board prevented the scraping blade from contacting the lower sections of the system. In other words, the system ran without the benefit of a lower constraining system, but only cleared the upper portions of the board.

To solve this problem we attempted to add weights to the bottom of the scraper, but without the benefit of a lower constraining mechanism, this introduced considerable pendulum at the outer extremes of the mechanism’s movement, where it had to change direction.

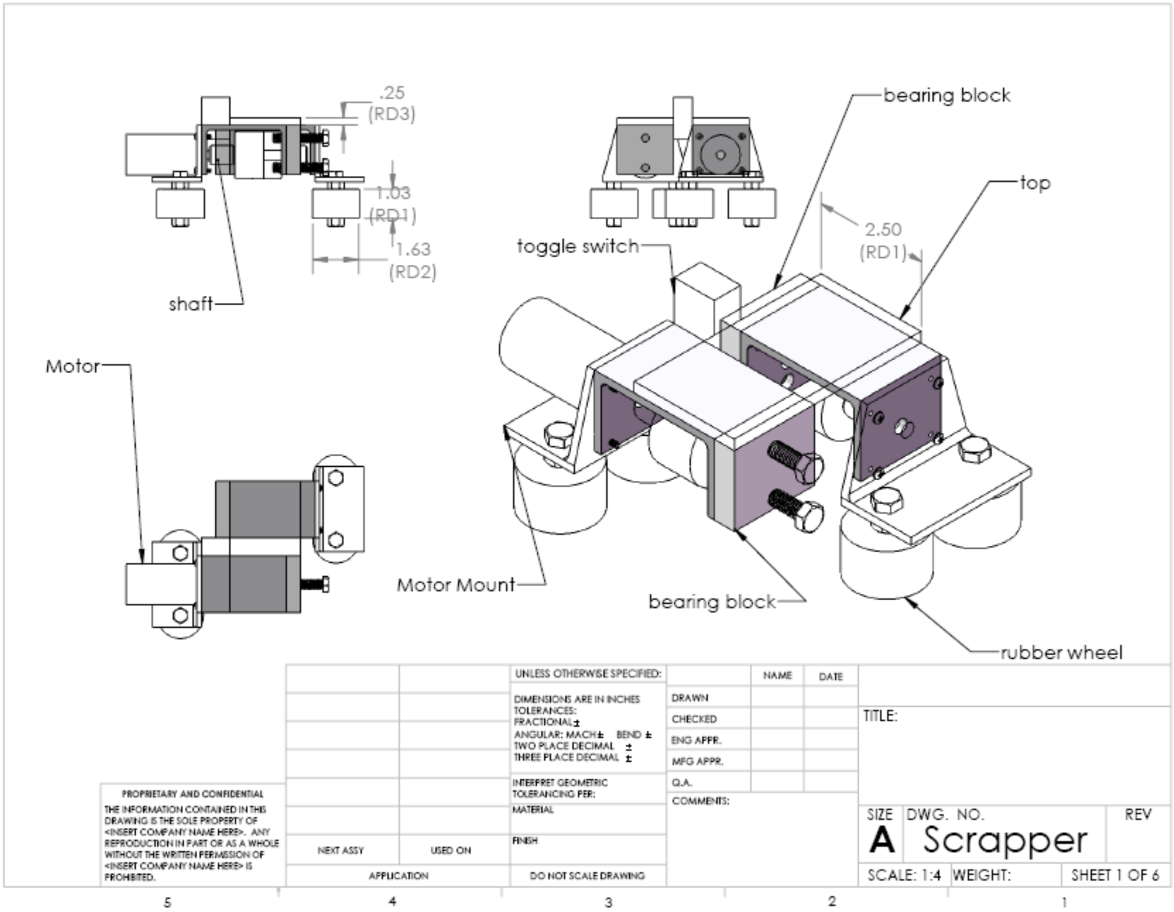
Once we had achieved full board scraping, we can to find that, due to an oddity of post-it note design, it was possible to crumple a post-it during the process in such a way as to jam the system. Post-it notes placed in the vertical position, with sticky side up, would be caught be the scraper just below the adhesive line, allowing the bottom of the note to fold while the adhesive became bound under the body of the scraper mechanism, halting the entire sweeping process. With the motor and voltages we were using, the system didn't have the torque to free itself from this position.

Figure 27: Post-It Note Jamming Problem



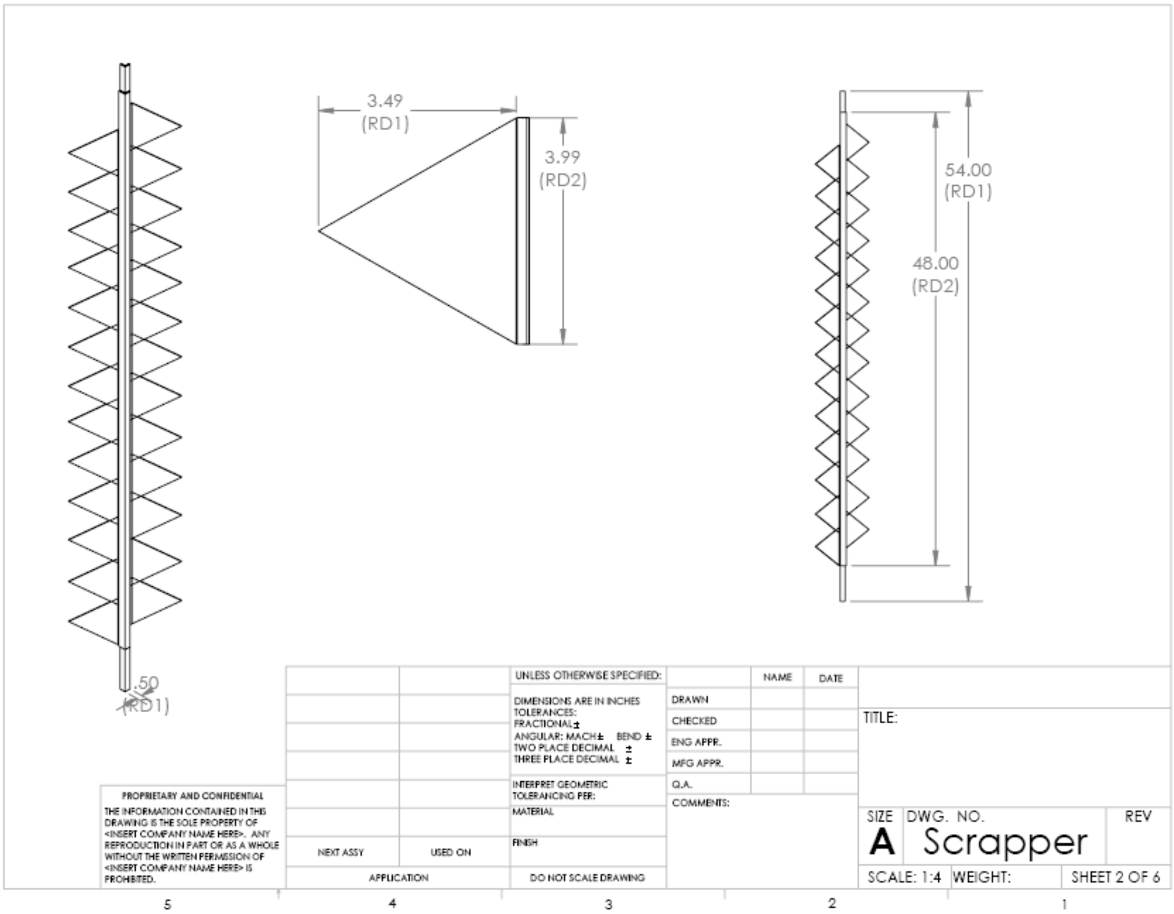
A number of solutions were proposed to deal with this problem, but it wasn't until the system was radically redesigned that we were able to make any headway against the problems we faced. To counteract the pendulum action experienced by the system, we attached the bottom bracket to the top (as seen in figure 28), giving the system two wheels in contact with the top of the board, as well as another two rolling wheels to constrain the forwards and backward motion of the top bracket.

Figure 28: Redesigned Bracket



Then we made significant changes to the sweeping blade. Long lengths of plastic were creating too much friction, and also causing the aforementioned crumple and bind problem. The redesigned geometry of the system allowed it to float above the board, just contacting it at the tips of the triangles. In addition to dramatically reducing the friction we had to deal with, it altered the way the blade would interact with post-its placed in the vertical upright configuration. Instead of binding just below the adhesive strip, the tips would either smoothly remove the adhesive line or, if it was positioned between the tips of one side, be undisturbed until the return path of the scraper, at which point the offset tips of the other side of the scraper would handle the situation.

Figure 29: Redesigned Sweeping Mechanism

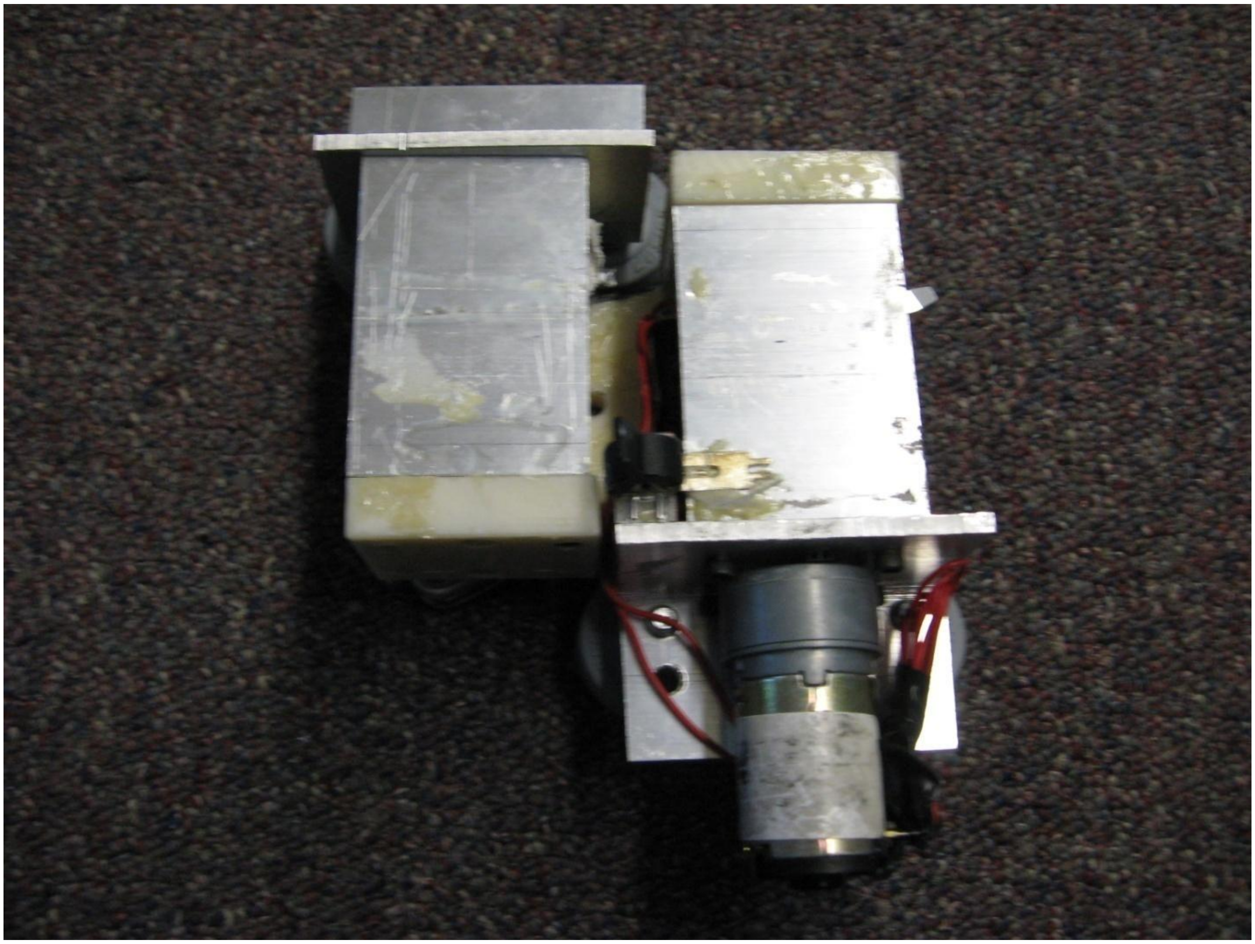


With these revisions we were effective able to remove spare populations of post-its from the board. However, large densities of post-its still could slow or stop the sweeping action. Our final revision was to increase the voltage of the motor driving the system. Originally we had used only 12 volts due to the constrains of our control board, however with the additional circuitry generated to enable 24 volts, the scrapper no longer became hung up when faced with dense regions.

2.5.5 Results

The end result of our revisions functioned well. Although one of the most troubling systems, requiring a number of design iterations, the end result performed admirably, and was capable of removing both the large clusters of post-its and the tricky solitary individual post-it indiscriminately.

Figure 30: Constructed Bracket



2.6 Switching Rig

2.6.1 Requirements

Due to numerous issues with our electrical control board, the decision was made to make the direction change of the sweeping robot entirely mechanical. This meant that the bracket on the top would have to travel the length of the board, reach the far end, and then have the voltages applied to opposite motor terminals, reversing the motor direction and driving it back the way it came, before repeating the same steps on the opposite end of the board.

2.6.2 Materials

- ABS ¼” plastic sheet
- 4-40 machine screws

2.6.3 Tools

- Machining mill
- band saw

2.6.4 Process

In order to accomplish this task mechanically, we created an effective H-bride with a double pole double throw toggle switch.

Figure 31: switching H-bridge

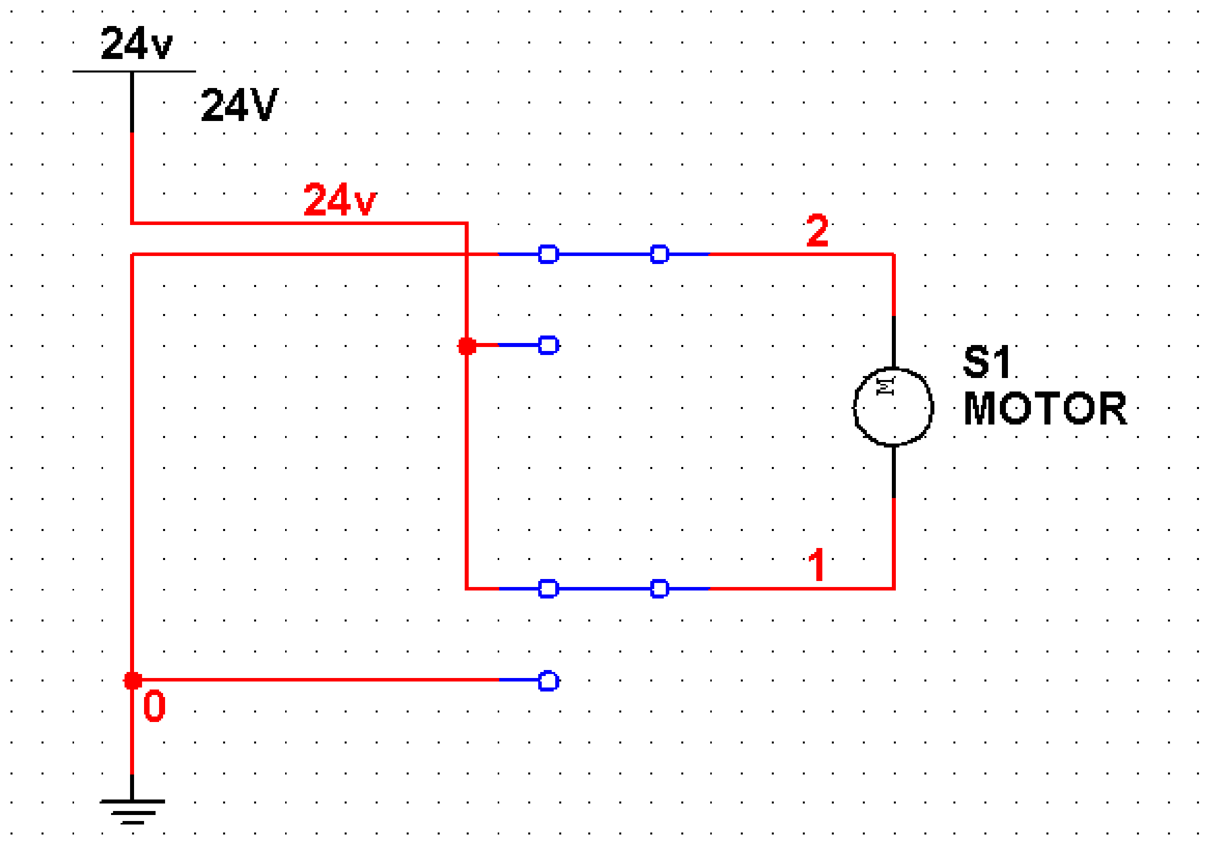
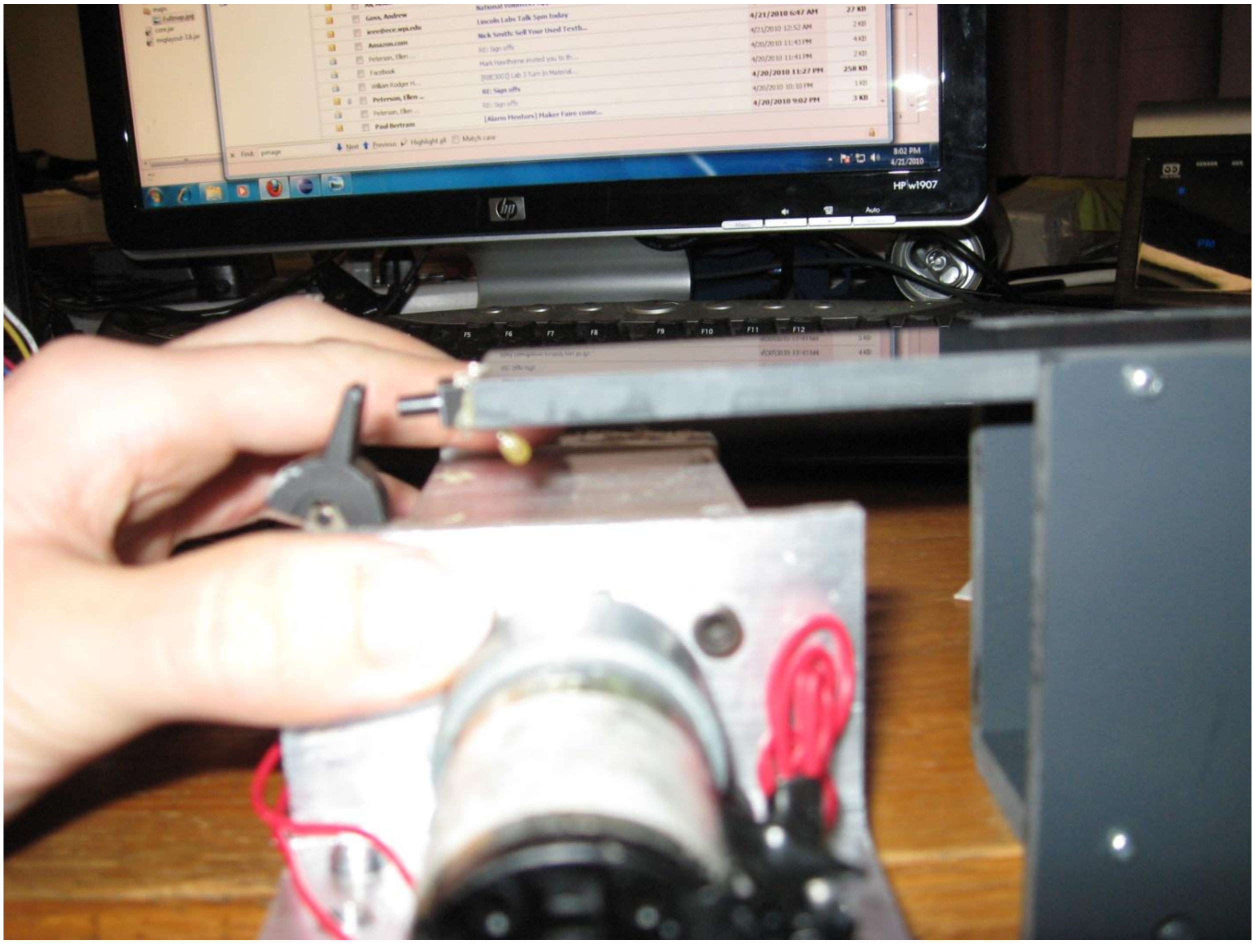


Figure 32: toggle switch

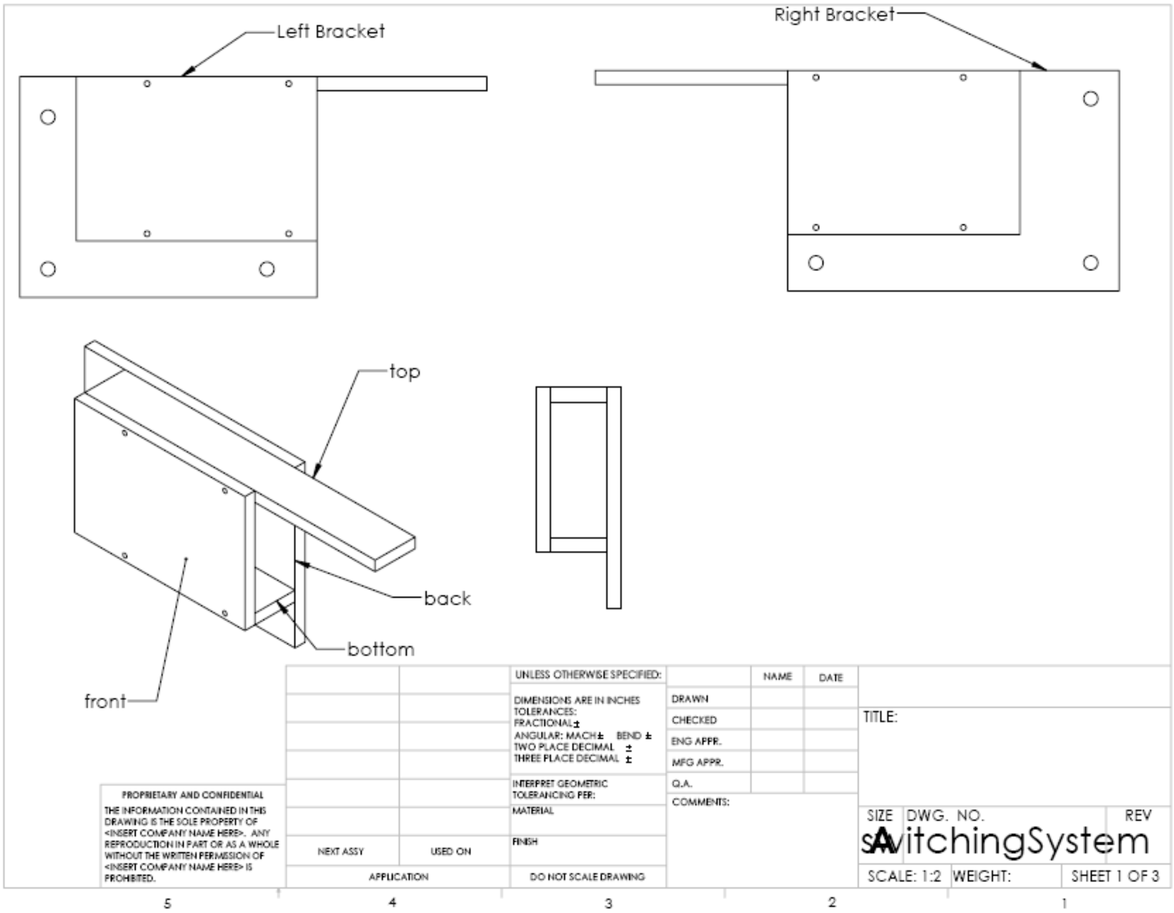
We were able to determine that using the kinetic energy of the system itself, we would be able to trigger a switch similar to the one pictured above. Such a setup however required a particular geometry however, to reach over the body of the upper bracket and contact the switch correctly.

Figure 33: Switching System in Action



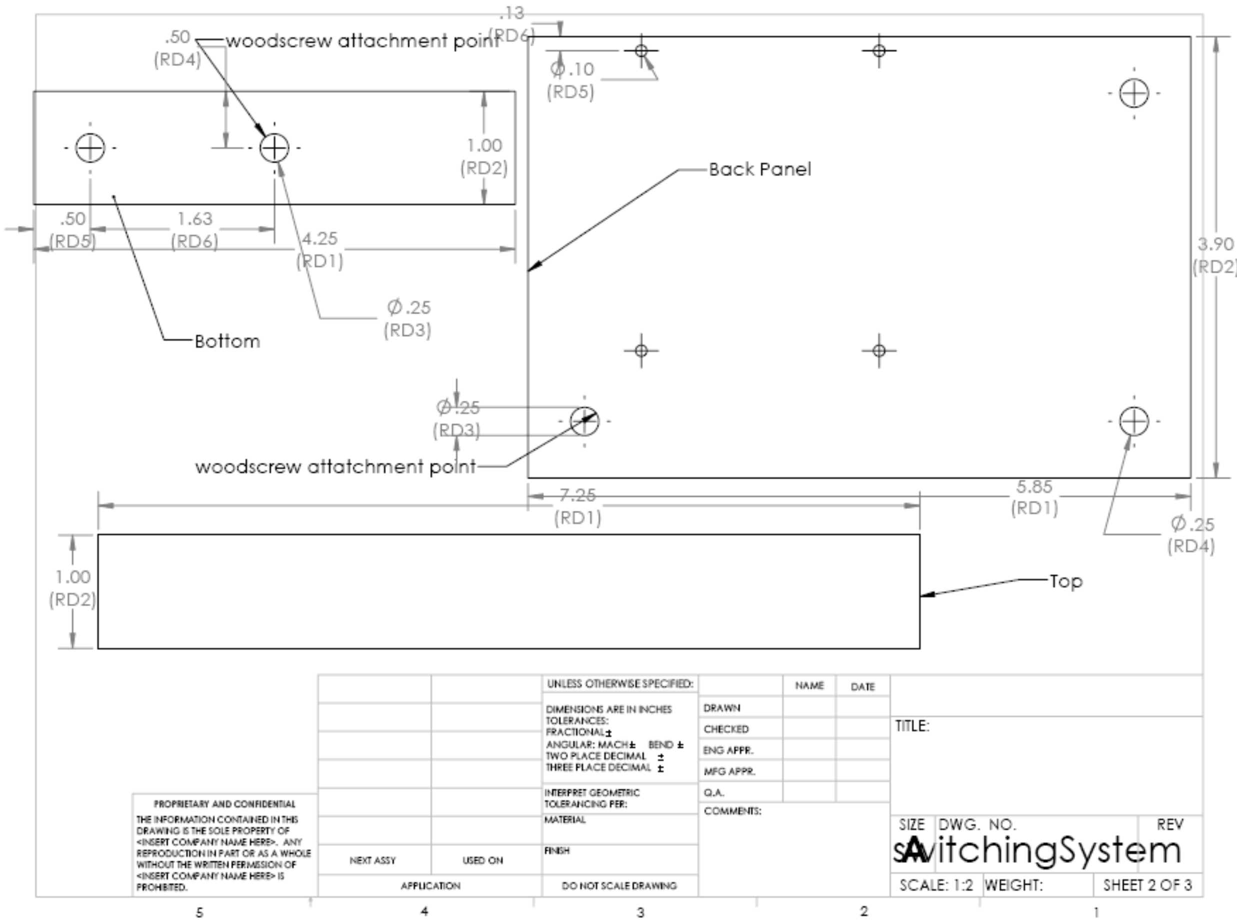
The solution to this issue was not difficult. Essentially we create a plastic box for the left and right sides of the board whose tops extended several inches beyond its sides. The internals of the boxes were utilized for housing other additional circuitry, and the boxes attached to the board via wood screws.

Figure 34: Explanation of Box Structure



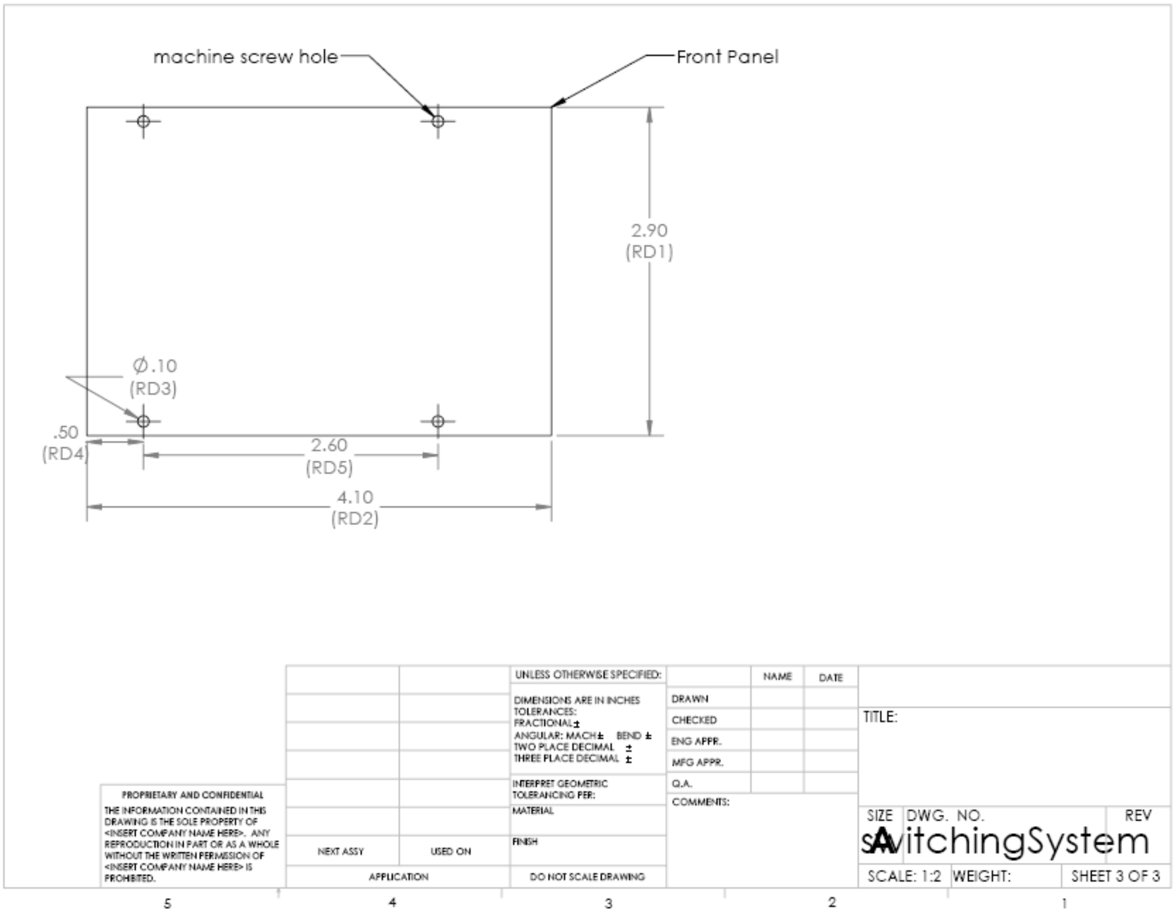
Connection points took the form of quarter inch holes on the bottom and back of the boxes, as pictured in figure 34.

Figure 35: Connection Points



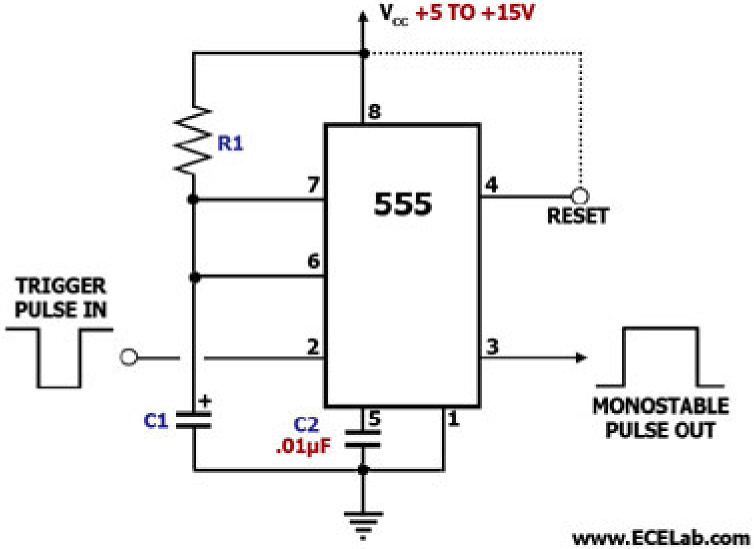
The internal of the box were made accessible by a removable front plate, attached to the rest of the system by machine screws (figure 35). This allowed access to the screw holes on the bottom of the box, as well as the circuitry held within, for the purposes of debugging and testing.

Figure 36: Removable Front Plate



Onto the left side was attached a momentary switch connected to a standard monostable 555 timer circuit. When the toggle switch contacted the momentary switch, the 555 timer held the output voltage high for several seconds so that it could be read in by our control board. This circuit helped prevent denounce and accounted for polling speed issues in software

Figure 37: typical monostable 555 timer



2.6.5 Results

The concept of a mechanical switching system was surprisingly robust. The sweeping system reliably changes direction as intended.

2.7 Software

2.7.1 Intro

From the beginning, we had agreed to work on this project in a modular fashion, keeping individual pieces separate so they could function independently. With this in mind, the software running this installation was divided into two separate pieces so that each could function on their own. The two main systems of the installation that needed to be software driven were the audio generation and the driving of the webcam and stepper motors. As such, we chose to code these as two separate systems, and later attempted some very simple interactions between them.

2.8 Audio Generation

2.8.1 Overview

The music generation section of this system was critical to creating an enjoyable interactive experience for the users. From the start, we intended on creating multiple different sets of music generation to both keep users interest and to convey different themes. We also were going through iterative development and wanted to be able to constantly be updating, changing, and adding new soundscapes. Because of all these reasons, we chose to keep the generation software as general and flexible as possible. For this, we chose to divide up the sound system into two individual parts – A system which interprets visual data and produces note values, and a system which takes in the note values and produces audio.

Luckily, the MIDI protocol already set up a perfect way for us to pass data between these two systems. MIDI is already integrated into Java, and there are a variety of programs which run synthesizers off of MIDI input data. We quickly agreed that we should use MIDI for this role, and kept it in mind when picking environments for these two systems.

To bridge the two systems, we determined a standard for the musical voices from the beginning. We chose to go with a simple setup of four main voices: A melody, a harmony, a drone, and possibly a sample bank. Using this as a standard, we created a layer of abstraction so that we could start working on either of the systems independently, knowing that they would meet in the middle and interact correctly.

2.8.2 Visual Capture and Interpretation

For the role of capturing video and interpreting the data, we ended up deciding to use the Processing[10] libraries. Processing is mostly built for image manipulation and as such was a

[10] http://processing.org/

strong choice for the interpretation we would be doing. There are also many libraries around that can be plugged into Processing to allow capturing images from a webcam, so we were able to use it for both of these jobs. We were also able to import the Processing libraries into Eclipse, instead of using their IDE, and use it as a standard Java library. This allowed us the flexibility to use it as we saw fit, as well as allowing us to use any other Java libraries or packages we thought we'd need. Access to other Java libraries was important, as it kept our implementation flexible and extensible; Java already had built in MIDI support, making the note value output relatively simple and kept it possible to implement communication with the driving system if we decided we needed it.

2.8.3 Capture

For the webcam plugin, we decided to use GSVideo[11]. It was fairly simple, and worked with all the webcams we were trying. Some of the other plugins we tried gave us trouble when we tried to use them through Eclipse, but GSVideo worked when we added the folder of gstreamer .DLLs into the Eclipse project. After including these libraries and setting up the build path, we were able to get all of this fully functioning inside Eclipse, which was a major advantage.

With GSVideo plugged into Processing, we were able to easily get webcam data. The two of them took care of pulling images from the webcam every frame, and Processing allowed us to get color data at each pixel of the image. This allowed us to get all our input data - we just had to decide what to do with it.

2.8.4 Interpretation

From the beginning, we recognized that this would be a major portion of our project. There is no simple or straightforward way of manipulating color values into musical notes, especially ones which sound pleasant and resembling music as opposed to just a swarm of noises. We realized this meant that we would probably be constantly changing and updating the way we interpreted the music. On top of that, we also were hoping to allow for multiple themes

[11] http://users.design.ucla.edu/~acolubri/processing/gsvideo/home/

and interpretations. Together, we knew that this meant we would need to keep the specific interpretation separate from the process and inner-workings of processing the data.

To do this, we built a set of classes which would aid in the creation of specific implementations. We started by laying out a system which consisted four layers of objects. Each object would be plugged into other objects, or would plug into another object, or both. Each layer was implemented as an abstract class so that we had some predefined behavior, but the specifics of each object in the layer were allowed to be different.

2.8.5 Inputs

At the lowest level we had Input structures, which would contain our lowest level of data. This would be things like pixels and their color and position, or other things like constants. These would be independent of other values, and would take no inputs themselves.

Inputs only had three main behaviors – they could be triggered to update their value, they could return their value, and they kept track of a min and max value they could ever return. For example, a pixel input takes a position of the pixel you want it to represent. Each frame, we want to update its value. Calling the pixels Update function causes it to get the value of the corresponding pixel in the current frame. At any point in time that we want to use the data, we simply call the getValue function, which will return its color. The minimum valueit could return would be 0, and the maximum value would be 255 (the range of color values).

2.8.6 Modifiers

The second layer was what we called Modifiers, as their main purpose was to modify data in some way, and pass it on. These objects take a number of inputs, allowing other objects to be plugged into them. They themselves were also implementations of the Input layer, inheriting all of the base Input behavior. As such, they could be connected to each other, allowing us to perform multiple operations consecutively on pieces of data.

The main behavior of Modifiers was whatever function they were performing on their input data. Some examples of this were summing their inputs, averaging their inputs, or returning an input if it was higher or lower than another input (high or low pass). This is performed in the Update function, and the getValue function returns its result, keeping it the same as the way an Input would be used. The modifiers also had functionality written into them to figure out what their minimum and maximum values could be. In the case of a sum, it would simply ask each of its inputs for their min or max and add them all together. The only real difference being that the Modifiers Inputs had to be set, but this could be done at any point, keeping the system flexible.

2.8.7 Output Adapters

The third layer was what we called Output Adapters. The main purpose of these objects was to take an input, and convert to some type of range. Since all inputs had a min and max value, we could map this to a given range of values. The simplest, yet most used implementation was to simply scale the input range to the output range. These were also implementations of Input objects, so their values were calculated during Update and returned by getValue. Their ranges and inputs were set in a similar fashion to Modifers. These objects were mostly just used in order to produce which made sense in the MIDI world for pitch and velocity (0-127).

2.8.8 Outputs

The fourth and final layer was what we called Outputs. These were basically just objects which would be bound to particular MIDI channels (and therefore instruments), and would be used to write data to them. The only real implementation we used at this level was one which would handle writing MIDI pitches and velocities. It took two inputs, one for pitch and one for velocity, and would write them to the given MIDI channel.

The only behavior these objects really had was a write behavior. This would cause them to ask their inputs for their values, and write them to the MIDI channel. No error checking was done, as it was assumed that a known valid value would be passed, even if it was using an Output Adapter to scale it to valid data.

2.9 Centralization

Since we realized that we were going to be implementing multiple methods of interpretations, we decided to create a set of objects which would handle all the repetitive work. This would allow us to focus on the actual interpretation algorithm instead of making sure it would run continuously and was set up right from the beginning. This way also allowed us to fix any issues or make any changes to the overall functionality of the program in one location, and all the algorithms would immediately take these changes. This is following the well-known object-oriented principle of centralization. We did this with two major aspects of the code – the updating of the layered objects and the start up of the program.

2.9.1 Updating Centralization

We realized that when we made the layered objects, we were almost always updating them every frame. Every time we made an object, we would initialize it in one place and have to make sure we remembered to update it later. Once we ended up dealing with large numbers of objects, we would tend to miss one here and there. To resolve this, we implemented a simple Factory Pattern.

We created multiple factories – one for each object layer. Any time we wanted to create one of these objects, we would request one from the appropriate factory. We allowed ourselves an optional Boolean parameter to the request that would allow us to tell the factory not to automatically update it – for the rare cases in which we wanted to update it manually. The factory would create a new one, return it, and add the object to a list of objects unless it was told not to. Any time the factory was told to update, it would iterate through all the objects in its list and update them (or, in the case of a MIDI output, it would tell the output to write its data). This allowed us to update all the objects automatically through one call.

2.9.2 Start Up Centralization

One task we noticed we would be constantly repeating was the set up for the Processing library and the webcam capture. Each algorithm would need start up exactly the same, but interpret the webcam data. In case our approach would change, or we would decide to add some functionality, we wanted to keep this away from the specific algorithms themselves.

To do this we created an abstract wrapper class for our general functionality. The responsibility was originally to start up a Processing applet and start the webcam capturing. When we went to start our testing, it also allowed us to replace the webcam capture with a version that would load movie files, allowing us to try our algorithms on predictable test data. When we implemented the factory system mentioned above, we also moved all the factory updates to this wrapper class, allowing the updating process to be fully automated. We also later used this to implement a small form of communication with the Driving System across all of our algorithms.

This was implemented in a Template Pattern sort of fashion. The wrapper class had abstract Setup and Update functions that were called from in the corresponding Processing calls. Any other base logic that we would always be doing would be called before or after the abstract functions as needed. Any time we implemented an algorithm we would simply create a new object which inherited from this wrapper class and put our logic into these abstract functions. The rest was taken care of by the wrapper class.

2.9.3 Algorithms

With the vision data manipulation system created, the actual programs, or sets, which convert pixel information into midi notes had to be generated. Over the course of the project about three sets were created, each with a different methodology for converting the webcam footage into music. Most of the differences between the generated programs stem from when they were made, seeing as later sets utilized methods that became available during their creation.

Despite their differences, the sets shared a central methodology: map different red-green-blue colors to the defined instruments and use the varying color intensities of the moving camera to drive the instrument pitch outputs. From this the end user can see that different color post-it notes cause different instruments to make sounds and that the more there is of a single color in an area, the more pronounced the corresponding instrument will become. Additionally, to keep notes from being sent when the camera passed over the black board, a simple highpass filter was used to remove any values lower than a set luminance threshold.

The bulk of the differences between the sets came from how we chose to treat the erratic color data and from where on the recorded image we chose to sample values. For most of the sets, averaging was needed to smooth out the rapid changes from the end musical output. The amount of averaging performed varied depending on the type of instrument driven by that modifier-chain; for the drone instrument, small changes over time were desired to avoid uncharacteristic fast note variation during generation, so heavy averaging of values were made over a large span of time to reduce noise in the changing data stream.

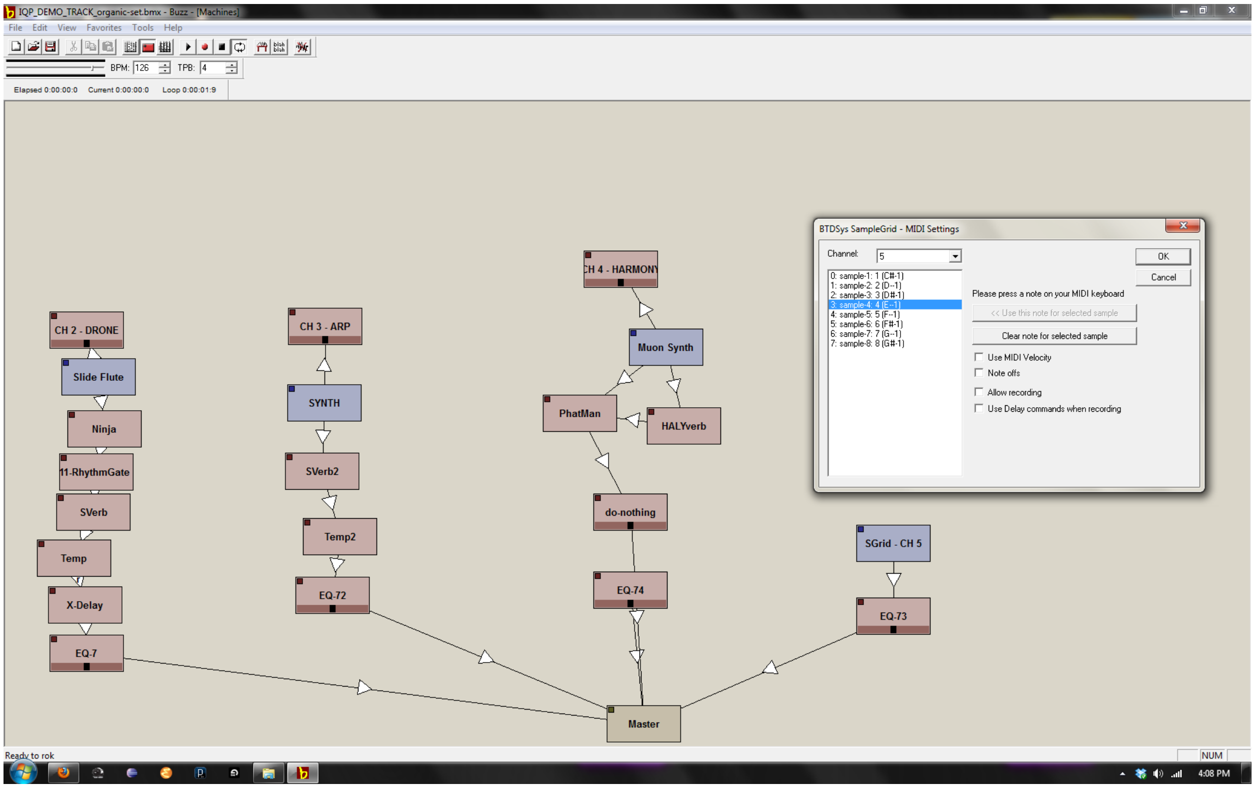
2.10 Audio Producer

For an Audio Producer, we had many choices of programs that we could use. We decided to use the MIDI protocol, so this brought us to a few major programs which we thought we could use. We weren't familiar with many, and cost played a major role in our choice. We quickly narrowed it down to Jeskola Buzz[12], which we ended up sticking with the whole time. It was free software and allowed plugging in new synthesizers and voices. We had a lot of control over all the voices and it allowed us to link different attributes of the synthesizers to MIDI channels as well, if we saw the need. One of us was already familiar very familiar with it as well, which was a major plus.

As stated before, we had initially laid out an interface of four voices. Using this, we set up multiple instrumentations. We made multiple Buzz set ups which consisted of synthesizers which corresponded to the voices laid out in our interface. We could load any set of these synthesizers at any point, allowing us to hot-swap voices as we saw fit.

[12] http://www.buzzmachines.com/

Figure 38: The Jeskola Buzz interface, showing a multi-instrument audio set and sample bank menu



2.10.1 Note Data Passing

The last piece we still needed was a way to move data from one program to the other. We had settled on the MIDI protocol, and found a program called MIDIYoke[13]. MIDIYoke basically emulates MIDI devices through software, and would pass any data written to an output back through an appropriate input. This allowed us to write to a MIDIYoke device from the Interpretation program, and it would be passed back out to Buzz.

2.10.2 Audio Generation Conclusion

[13] http://www.midiox.com/index.htm?http://www.midiox.com/myoke.htm

The approach we took to the Audio Generation worked out very well for us. It allowed our system to remain modular as we had originally intended. We were able to produce it in pieces and made sure they worked on their own. We were also able to implement different pieces in parallel and in different stages, knowing that we could bring it all together afterwards since we had a predefined interface laid out. We also managed to keep it centralized, mostly due to the Factory and Template approaches, so as our requirements, approaches, and ideas changed, our system was able to adapt.

2.11 Driving System

2.11.1 Overview

In order to keep the data changing, we needed some sort of system to move the webcam across the board. For hardware, we had already decided to use StepperBee[14] control boards to drive two stepper motors with a timing belt running between them. The webcam would be mounted to this belt, so we needed to implement a system to drive these motors to allow us to read all the data off of the board.

Although the original intention was only to drive the motors with this software, the StepperBee became our main interface to the physical world. Any other physical device such as the Scraper Bot or any lights, needed some sort of electrical interface. Since we were already using the StepperBees and they had digital outputs, these became the way we ran all these things.

The other software was also very passive and didn't need to know where it was or what the webcam was doing – just what colors it saw. As such, it made sense for us to put the rest of the event logic and user interaction handling in with the StepperBee side of the code. This section become the center of the software, and played a more authoritative role.

From a software standpoint, a lot of the functionality revolved around the StepperBee itself, so the lines between roles weren't as distinct. The system was set up into a few logical tasks, which were mostly independent of each other, but all still interacted with the StepperBee

[14] http://www.pc-control.co.uk/stepperbee_plus_info.htm

in general and needed some sort of authority and management. Broken down, these tasks were managing StepperBee interactions, handling where to go, and an overall management system.

2.11.2 StepperBee Management

The first task here was to set up the lowest level system which would handle controlling and managing the StepperBee. The products came with a .DLL with functions for simple operations like initialization, running a motor a certain number of steps, and getting the current status. To make this more suited for our own uses, we wrote a class around this DLL to keep track and handle the StepperBee.

The biggest objective of this class was to allow us to reference positions on the board as Cartesian points. The StepperBee DLL works entirely in just numbers of steps, which was not convenient to deal with at all. We wrote functions to move the camera to a given point, which would convert these points into a set number of steps using a simple algorithm based off of the Pythagorean Theorem. This allowed us to drive the camera in a fashion that was easy to think about and simple to comprehend. This class was therefore also responsible for keeping track of where the camera was.

In addition to these responsibilities, it was also responsible for handling any other board status. The main things here were the digital inputs and outputs on the board. These would be found every time we called the UpdateStatus function from the DLL. Since the class was wrapping the DLL, any time this update function was called, the class would keep track of any changes to the information. We set up simple properties on this class so that we could read inputs or write outputs very easily.

2.11.3 Movement Handling

We had two different systems for determining where to move the camera, but both ended up writing points to the StepperBee class we made. The first way we moved the camera was selected a random predefined pattern. Each pattern consisted of different lists of points for the camera to follow. We wrote a manager class that would take one of these patterns and feed it into a queue. It would ask the StepperBee if it had reached its target every 10 milliseconds or so. If it had, it would pop the next position off the queue, and repeat. Once the queue was empty, it would choose another pattern at random and add it to the queue.

The second way was user control using a joystick. We set up a simple class which used DirectInput to poll joystick axes and buttons. Every few milliseconds, it would check the X and Y axes from the joystick, and use them to modulate the current X and Y position of the StepperBee. If the StepperBee did not have a new target, it would pass this modified position as the new target and repeat. At each poll of the joystick, we would update the current position and only modify it by a small distance. This delayed the responsiveness of the camera movement a little bit, causing it to be a bit jerky, as it could only move when the StepperBee had reached its next position. There was no easy way to stop the motors and figure out how close to its target the StepperBee really was, so we figured this wasn't an issue really worth addressing.

These two systems would both send their desired targets to the StepperBee in the same way, allowing us to keep the system modular. They were both interfacing with the same code, so if we ended up fixing or updating something in one, it would affect both.

2.11.4 Overall Management

The third task that needed to be completed by this system was some form of overall management. Pieces like the actual locomotion of the webcam and how to turn on and off inputs were set up, but we needed a way to determine when to perform each of these actions. This drove the need for an authoritative management system.

The main goal of this system was to handle the events that we had decided we were going to have. We wanted to have a set period of time where the installation would be idle, allowing people to come up and interact with it. After this period of time, the installation would begin reading the data on the board for a certain period of time. It would then stop and repeat the process. This mostly consisted of two large timers that were used as transitions between the two main states – reading and sitting idle. The idle state was very simple – the installation would just wait for the timer, then trigger the next read state.

The read state would activate the appropriate movement handler, whether it was joystick or autonomous control. This resulted in a hierarchical state machine – the read state has its own state machine inside of it. The initial state was defined to be autonomous control, but from there it would check if a joystick button was pressed by the user to request control. If a user had requested joystick control, this management system would tell the autonomous control to stop when it could, as it could not stop during a movement. When the autonomous control said it had successfully stopped, the management system would turn over to the joystick control and let the user do all the movement. This whole reading state would proceed until the timer expired, and would then move to the idle state.

We later added a few things into the transitions between the idle and read states. When going from idle to a reading state, we would pause for a few seconds and flash a siren mounted to the installation to notify users that it was about to start moving. On the transition from reading to idle, the installation would sometimes run the Scraper Bot to clean up the board.

We also investigated setting up a UDP connection between this management system and the Audio Generation system. We wanted to be able to switch sets of voices and which audio interpretation algorithm we were using, either randomly between events or through user control. Due to time constraints, this was never fully fleshed out, though we had a simple version working. We would send signals over UDP informing the Audio Generation system to mute or un-mute everything. This message would be sent at the beginning and end of events as appropriate. The Audio Generation system would catch this message, and pass a MIDI message along to Buzz which would turn the volume up or down. The system was in place to do more with it, but we weren't able to fully implement all our ideas.

2.11.5 Driving System Conclusion

We managed to succeed in keeping our system modular here as well. We allowed for changes and updates to be made to the system. This system was mostly divided into three pieces, each of which only depending on small portions of the ones around it. For example, the handling systems only needed to receive start and stop signals from the management class. This allowed us to greatly change the logic in any of the three portions of the system without breaking anything in the others. As our project shifted throughout the course of the year, the system was able to shift and change as well.

3 Results

3.1 Museum of Science Experience

On March 16th an open call for robotic musical installations or related projects was made on the DorkBot Boston mailing list. After a lengthy correspondence with an exhibit coordinator we were invited to install and display our project to the museum-going public. After two weeks of intensive work our project was displayed at the Boston Museum of Science from April 11 to April 17 2010. It was during the Robot Block Party event at the MOS. We went into all of this really excited because this gave us our golden opportunity to show not only our families what we have spent our whole year working on but to see other people interact with our installation which had been the entire point of our project.

However things did not work out as we planned when we heard we were going to the MOS. Our group was put through some trials before we got out project successfully working. The first obstacle we ran into was that our scraper-bot, which was supposed to clear all the post-it notes off the board when it was full, was underpowered. So as a group we worked late nights to fix the problem, which involved a redesign of the power circuit to deliver 24 volts instead of 12 volts. We quickly realized that our computer power supply was not able to output enough current to power 24V to the motor and we had to rethink the system. So in a last minute scramble a new circuit was designed and implemented and that ended up being the circuit that worked for us. This involved at 24V AC power supply bought Saturday morning right when Radio Shack opened their doors and it was integrated with a rectifying circuit to make it 24V DC into our board. The result was a success and we all left our project that day happily looking forward to tomorrow. However we ran into our second pitfall the next day.

The next day our installation was transported to the MOS and then we ended up spending most of the day setting it up. It took us almost all day because our camera automation was not working correctly. This second problem tested our group more than the first because we had our families all watching as we were trying and failing for hours trying to get our project set-up and running. In the end with some debugging and a lot of code review we managed to get it working enough to allow people to interact with it. We had half the group checking and rechecking all the wiring while the other half checked the software. However by the time we got it working it was late in the day and most of the functionality of the board was down. As a result viewers really did not get a good idea of what was going. So at the end of the day we devised a plan to spend the week writing more code and making adjustments so that when we came back the next weekend we would have another day to show people our project.

When we came back that next weekend on Saturday people got to really experience our project. We ended up scrapping the buggy self-automated camera scan code because the stepper motor control board we were using was losing the information we were sending it. This means that our camera would lose its way and eventually end up not properly scanning the board. So instead we gave that control to the spectators. People came up to the board and not only put a post-it note on the board but now used the game controller we had to control the camera and therefore control the music as well. After we got this all set-up about the first hour and a half we allowed people to come up and start interacting with our installation. In the end people liked and enjoyed our installation, especially the kids.

Figure 39: MOS Experience 3

Figure 40: MOS Experience 4

People who put a post-it note on the board were delighted to pass the camera over their post-it note and hear the music that their contribution to the board created. We even encouraged the kids to write their name on their post-it so they would know which one was theirs. This was the most enjoyable part of the entire experience for everyone in our group. This was the moment where we said to ourselves “this is what this whole year was about.” It was seeing something that we created for people to interact with actually working, and people being thoroughly amused at what music they could create with just some colored post-it notes on a board. We even saw a few of the same people and their kids come back for another try at our installation so they could play their post-it note.

Figure 41: MOS Experience 1

Figure 42: MOS Experience 2

In the end after a few rocky starts with our installation between the scraper bot and the automation of the camera our project worked out even better than we expected. The camera automation that we had to scrap allowed the user to have a more intimate experience with our installation than they would have otherwise experienced if everything worked as it should have. The changes that we made to our installation made our project a better experience for people at the MOS.

3.2 Project Postmortem

In many ways our project performed better than expected. Although the audience we ultimately achieved at the Museum of Science was not of the predicted demographic, our system performed admirably. The board, although designed for full grown adults, was still easily accessible by younger users, who put them on the lower sections of the board. The higher sections of the board were left for the parents of small children, who would often assist toddlers in placing post-it notes. Difficulties with the control boards had prevented the automation of the sweeping process and demanded periodic manual resetting of the system to account for lost stepper motor steps. However, this proved to be a truly insignificant incontinence since the quantity of users never exceeded the capacity of our board space. The joystick control in particular was very successful in guaranteeing the involvement of youths too small to interact fully with the scale of our board. Our time at the Museum of Science showed that even without a fully automated system, we could engage individuals and bring people together in the act of creating music.

Despite these successes, this project faced a number of indisputable failures and setbacks, all of which are potential lessons to future iterations of this, or similar projects. The most glaring issue that was faced by our development team was the lack of time. In a project such as this, with the development of significant physical components, it is easy to fall behind schedule in the development process. Once behind schedule, the natural iterative process of prototyping and redesigning cannot be adhered to, and component subsystems suddenly begin to fail to meet design criteria.

For our team, several factors can be attributed with the delays we experienced. The first was a lack of well defined subsystem interfaces. Because of the very abstract and conceptual nature of our project, many decisions regarding the final form of our installation were repeatedly pushed back. This resulted in difficulty defining precisely what any one system would do, or what constraints it would operate within. In other words we were forced to make design decisions without knowing all the facts. This problem could easily be solved with more rigorous design iterations and greater full team involvement in design reviews.

Similarly, issues with inter-team communications often stalled project development and concealed fatal flaws. Very often the mechanical sub team would have a particular solution in mind, but the software team would be uninformed of the necessary requirements until weeks later, or vice versa. This resulted in the rushed combination of subsystems in the final weeks of development, which revealed a number of issues too late to rectify properly. Several glaring solutions exist to this particular problem. The simplest would be to redesign the leadership hierarchy of the project. While, in our definition of member roles, there were individuals whose duty it was to record meeting data, this role must be assumed by all members of the team. The utilization of shared file systems in the form of dropbox and gmail accounts certainly contributed an air of transparency across sub groups, but without proper documentation, these resources often were too convoluted to be of any real help. Perhaps a stricter adherence to the practice of posting weblog updates with layman explanations would result in better, simpler, and more accessible documentation. Alternatively, a well defined team leader in future project iterations could ensure that every sub team was able to communicate effectively with the others.

The final issue of our project turned out to be the difficulty in resource obtainment. Although many of the resources for the fabrication demanded by our project existed on campus, we faced significant resistance in obtaining machine time. In future iterations of the project, efforts can certainly be made to minimize the existence of fabricated parts, thus removing this issue altogether. This issue may also have been avoided with more educated predictions of project requirements. If we had had a clearer concept of what would be required by our installation, we could have begun exploring the proper channels well before critical deadlines in our design process. In essence this is a result of the first issue mentioned, the lack of well defined system requirements.

The overarching message should be clear. The most beneficial course of action for our team would have been the selection of a final design much earlier into the project. This would have allowed for better system designs, the timelier acquisition of critical resources, and a more relaxed development cycle which would have allowed us the time necessary to better document our progress as we made it. As in any project, there is a shaky balance between the time spent planning, and the time spent executing a plan, and the hardships we faced with this installation show that we tended to close to the former.

As for correcting the technical issues we faced, the first act of any group attempting to continue this project should be to phase out the StepperBee boards. These pieces of hardware were undoubtedly the weak link in our project. They behaved erratically and performed inconsistently. We found the documentation to be limited and labels to be confusing. There was a mutual consensus among the team members that if we were to do this project again, we would use regular DC motors with some form of feedback instead (most likely a pair of rotational encoders). This would force us to develop our own control mechanism for this closed loop system, but being able to know the position of the camera absolutely would make up for the time spent developing this system. The decision to switch from stepper motor to DC motor would simply be a cost saving mechanism. DC motors of equivalent torque are comparatively much less expensive. Moreover, using a DC motor would make it easier to interface the motors with whatever we used as a control board. Essentially, the whole system would become a microcontroller being passed position data from the computer through a serial line, the microcontroller generating the necessary set points for the position, and then controlling the motors through some form of H-bridge circuit in a continuous PID control loop. This would solve the lost step issues that was causing the migrating of the camera's home position (which in turn required the manual resetting of the system periodically), and would have the added benefit of allowing for reliable I/O for the other systems (sweeper, warning light, inputs, etc.) which we never could achieve on the stepperBee.

In the Appendix A section of this paper is some of the work that we did this year that shows some of the changes that we went through. There is a copy of our proposal[15] that we wrote up to get an idea of what we wanted to achieve as a group. There are a couple of artist rendition[16] pictures in there that express what we thought we were going to do. Obviously there were some changes. For instance also in the Appendix A is a section that has our construction pictures[17] of us building that board and another section that has the potential names[18] that we thinking about calling our project.

---

[15] See Appendix A section 7.3
[16] See Appendix A section 7.3
[17] See Appendix A section 7.2
[18] See Appendix A section 7.4

For the longest time we called our project SaraSong which was a Buddhist reference so the cycle of life and death because that is what our music was acting it was lively when the board is full of colorful post-it notes and then dead when the board it wiped clean for the next cycle to begin.

Another key change in our project was for most of the project design we wanted to implement the use of thermal paper to use a the medium for people to come up and draw on and then later a robot would come by and heat up the paper and black out all the color that people put on the board and that would be our music creation cycle. Instead we changed to post-it notes for many reasons from simplicity because we did not have time to build everything necessary to be able to use the thermal paper. Also the thermal paper was expensive and out of our budget which is why we sent a proposal to the Awesome Foundation[19] to get funding for the project. We did not end up winning that unfortunately. The last reason we did not go with the thermal paper was the thermal paper has issues working with markers as the solution that is heat activated is water soluble and when the markets marked up the paper they removed that solution and so the colors would not be blacked out as we intended. However in the end post-it notes worked out for the best and people enjoyed them and they where a good solution to the problems we ran into with the thermal paper.

4 Conclusion

The main goal of our project was to design and implement an installation which served some sort of purpose. Early on, we decided that this purpose would be bringing people together to work towards one common goal. We wanted to use technology in a way that would get people to interact in person and away from the impersonal sense that we generally see today with things like the internet. As proved by our Museum of Science experience, we were successful in doing so. Though it wasn’t exactly our intended audience, people of different age groups and people who didn’t know each other would come to the board and interact with our installation together. We had hoped for more person to person interaction as well, but we did manage to break the ice between certain individuals and we succeeded in getting people to interact with both our installation and each other.

---

[19] See Appendix A section 7.1

Another major goal of this project was to keep the system modular and flexible. This was a goal we met in some places and not in others. We looked at this more along the lines of a plan for implementation, though to an extent we overlooked the modularity of the final installation. Throughout the year while we were creating the installation, our goals were constantly changing, as we had expected. For the most part, our project and final product were able to adapt as we went, taking in new pieces and throwing out others. Though this worked during the term, when things went wrong with the final installation, we didn't have easy ways to cut pieces out. We didn't predict having problems with our final product when presenting it, so it took us some work to get everything working when we went to present it. Though it wasn't as easy as it could have been, we were still able to get our installation working in a reasonable amount of time, just not as quickly or as well as we would have hoped. This is something we wished we had foreseen better, but it still worked out for us, and the project was still a success.

All in all, we managed in meeting the goals that were initially laid out to us from the beginning of the term. Our shortcomings laid in the goals that we set out and hoped to achieve on our own, though these were smaller and not critical to the project's functionality. We succeeded in our initial intentions, and produced an operational installation which managed to bring people together, deeming this project a success.

5 External source appendix:

StepperBee + Product Page:

http://www.pc-control.co.uk/stepperbee_plus_info.htm

IPA IQP Project Blog:

www.ipa-iqp.blogspot.com

Lucky Dragon “make a baby”:

http://www.notthisorthat.com/sblog/index.php?/archives/94-LUCKY-DRAGONS-make-a-baby.html

Cybraphon:

http://cybraphon.com/

6 Appendix A

6.1 The Awesome Foundation Application

Project Proposal

SaraSong Interactive Installation

An Interactive Qualifying Project Proposal for the

Interactive Public Art Team:

Christopher Earley, Dylan James, Nicholas Smith, Seth Crocker

Project Advisor: Professor Joshua Rosenstock

Worcester Polytechnic Institute

2009-2010

Summary:

SaraSong is an interactive installation that uses color-pencil pictures drawn by participants on a special robotically-augmented paper canvas to create live visual and auditory compositions that reflect on the cyclical nature of life on earth.

Its name, "SaraSong", is a play on the word sangsara (or samsara) which in Buddhism is the term used to describe the endless cycle of birth, death, and rebirth that all things are governed by. In this installation, the elements that will undergo the transformative cycle will be visual and musical. Taking inspiration from the Buddhist ritual of creating then ceremonially destroying complex sand artworks, called mandalas, sarasong will ask for input from the public in the form of drawings made on a thermally-reactive paper covered surface, which will then be read and converted into music by a wall-mounted robotic scanner.

As time progresses, there will be ceremonial blacking out, or deletion of the contributed scribbles by another robot that exposes the thermal-paper surface to heat. This removal of input will serve as a proxy the final outcome of time's movement, forcing the information of the past to pass into a state of non-being. But although the user contributions are forever gone visually, audibly their influence and characteristics will endure in the computer generated soundscape for the operating lifetime of the installation. This seeks to mirror the duality of mans' passage through time; although physical permanence is an impossibility for any being, it is wholly possible for the interactions and influences of man to persist indefinitely within the physical realm.

Description:

The Installation consists of five major parts: the wall-mounted thermal paper drawing surface, a wall-mounted vertical plotter used for reading the contributed drawings, another vertical plotter that exposes the paper to a stream of heated air, the computer system that coordinates the movement of the robots and the flow of installation data, and finally the media generation algorithm that create the additive audio-visual output of the installation.

The drawing surface that users will mark on is made up of a wall-mounted butcher paper roll dispenser equipped with a large roll of thermal paper and some clips to keep the free end of the paper against the wall. The paper, which is functionally identical to the paper used for printing receipts, will allow the color-pencil drawings from the users to be read by the first plotter, but after the paper is exposed the heat, it immediately turns black, effectively making any attempt by the plotter to get useful info from that section of paper impossible.

The apparatuses that will be performing the reading and thermal exposure tasks are mechanically similar; both are simplified vertical plotters that move along the drawing surface through the use of digitally controlled electric motors and both carry a small payload of equipment that allows them to accomplish their tasks, but that is where the similarities end.

The first plotter, the ‘read head’, uses two stepper motors, mounted to the top corners of the drawing surface, and toothed belts, connected to the read housing, to pull the payload to any point on the drawing surface. The payload that will be used to grab color information from the drawing surface consists of an off-the-shelf webcam, for high-definition pictures of the user drawings, and a small array of white LEDs, to smooth out the variations in lighting that might skew the colors of the pictures being taken by the webcam. This manner of two dimensional movement has been heavily utilized by other digital installations, such as hektor and viktor, as a way to create a cheap manageable device that is capable of drawing over large surfaces, but to the best of our knowledge, this style of vertical plotter has never been used for input.

The second vertical plotter, the ‘death head’, uses a toothed bar and a gear to facilitate horizontal movement and a high torque winch to move the payload vertically over the drawing surface. The payload for this plotter is made up of a cheap heat gun and a special nozzle that allows the stream of hot air to flow from the neck of the gun at ninety degrees; this was needed to diminish the amount the payload sticks out from the wall. The speed in which the motor assembly moves the payload will give the paper ample time to become exposed to the heated air and fully change color.

Controlling all of these elements will be a mid-range laptop equipped with Windows XP, this was done due to linux’s inability to handle the easy transfer of media/midi data and for stability reasons, and running a host of free and custom applications. In its essence, the computer system will coordinate when the participants can populate the drawing surface and when the plotters take over with the ritual reading and destruction of the user contributions.

In the performance stage, after the users have filled the paper with drawing, The ‘read head’ plotter follows one hard-coded path over the drawing surface, taking pictures at a set interval and feeding those images to the computer for color and shape analysis. As the read head progresses, portions of the surface will be blacked-out by the ‘delete head’ plotter, routine will be pre-programmed to remove the possibility of collision between the plotters.

After the plotter routines are completed, the participants will be allowed to add more drawing to the surface. During this stage the information collected from the previous readings will be used to generate auditory and visual compositions that owe their creation from the visual input from previous audiences.

The algorithms that will be generating the visual and musical output of the installation take inspiration from the systems that they aim to portray: the complex interactions of living things. Now the number of interacting agents in this system will be low, making the resultant output patterns a gross, but simplification of the sheer intricacy that is inherit in actual interpersonal interaction.

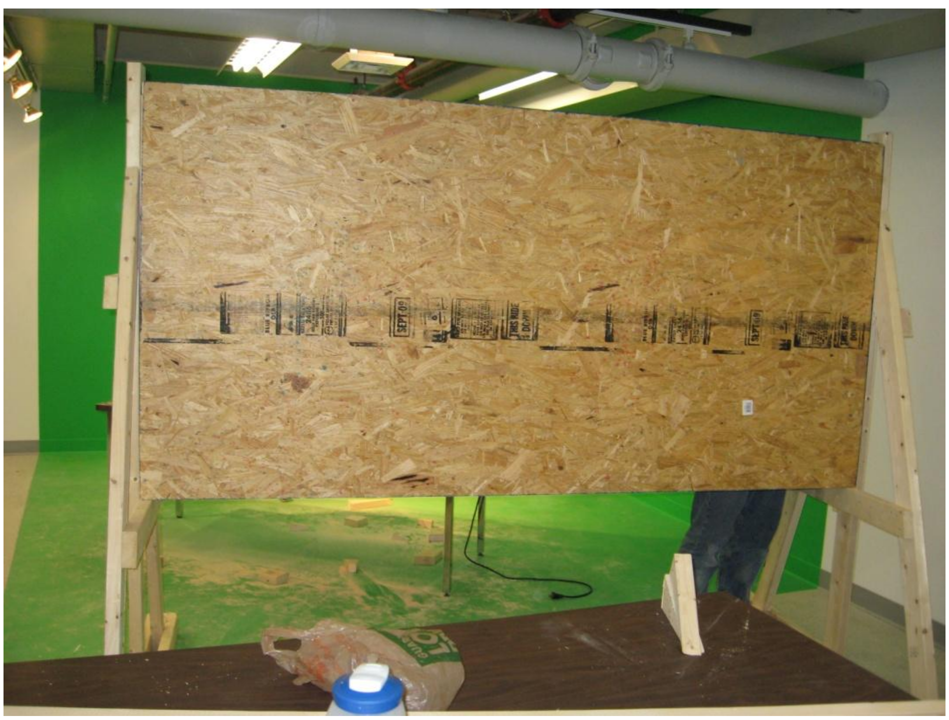
The image data that has been processed by the computer will feed its color and position data into the networked system of interacting agents and from that seed, a complex and continual data stream will be created from that data and the resultant inter-agent communication. This data is used to generate the movement, frequency, and timbre of the pre-generated musical elements that make up the auditory composition.

For visuals, a collage composed of a literal representation of the node-and-link complex network and the processed images from the ‘read head’ plotter will be generated and displayed on the screen of the laptop computer in real-time.

About Us:

The Interactive Public Art Team is a multidisciplinary group of four students from Worcester Polytechnic Institute chosen to design and implement an interactive installation that utilizes user input and facilitates a dialog between the user and the system that is mutually advantageous for both parties. Our hope is to create an experience that is engaging and thought provoking for the users while, at the same time, the information supplied by users allows the installation to continue operation and generate output.

6.2 Construction Pictures



6.3 Project Proposal

Proposal for "SaraSong" (http://en.wikipedia.org/wiki/Samsara_(Buddhism))

introduction

inspiration

how this idea came around

monks - sand arts

name description

precedence

intro sentance

Its name, "SaraSong", is a play on the word sangsara (or samsara) which in Buddhism and Hinduism is the term used to describe the endless cycle of birth, death, and rebirth that all things are governed by. In this installation, the elements that will undergo the transformative cycle will be visual and musical. Taking inspiration from the Buddhist ritual of creating then ceremonially destroying complex sand artworks, called mandalas, sarasong will ask for input from the public in the form of drawings made on a thermally-reactive paper covered surface, which will then be read and converted into music by a wall-mounted robotic scanner. As time progresses, there will be ceremonial blacking out, or deletion of the contributed scribbles by another robot that exposes the thermal-paper surface to heat. This removal of input will serve as the final outcome of time's movement, forcing the information of the past to pass into a state of non-being. But although the user contributions are forever gone visually, audibly their influence and characteristics will endure in the computer generated soundscape for the operating lifetime of the installation. This seeks to mirror the duality of mans' passage through time; although physical permanence is an impossibility for any being, it is wholly possible for the interactions and influences of man to persist indefinitely within the physical realm.

transition

small tech overview

precedence

theme outline

creating/destruction

art vs. authority

futility of life

Samsara

Brief Intro/Transition...

One of the major themes present in SaraSong is the theme of creation versus destruction. The installation allows users to come up and add something to the wall. The read head will eventually come by, possibly scan over that area, and create music based off of the users addition. Not only is the user creating their own piece of art, but the installation itself creates music based off of the users interaction with the wall.

Eventually, a second head may come by and blot out this users piece, destroying what they had created. The user's piece may have had a lasting effect on the characteristics of the music, allowing a piece of them to survive. There may be small remains left over afterward, but the substantial part of the users piece is, for the most part, destroyed. This also resembles the theme of the futility of life and sustenance of actions; Things are brought into this world, but are eventually destroyed, leaving only a small physical trace behind. Though their physical entities are destroyed, their interactions and creations can and will be sustained by the things around them.

These interactions also reflect the buddhist idea of Samsara, or "continuous flowing". Things are constantly being brought into the world through birth and taken out through death. The Buddhists believe that only through the process of enlightenment, and achievement of nirvana, can anything escape the Samasara cycle. The physical pieces on the wall are constantly being applied by users, and removed by the destruction head, stuck in a constant cycle between creation and destruction. Despite this cycle, their effects on the music can and will pass out of the cycle, and into a more permanent state than could have ever been achieved physically - in the effects on the music.

This destructive head can also be seen as an authority figure, regulating the creative interactions made by the users interaction [from user's perspective]

narrative

concept art

episodic performance [let users draw and then read]

On a brisk Sunday morning walking along the park path you notice a wall you've never seen before. Surprising, since its size is not something that would typically allow it to be missed. Its maybe 8 or 10 feet tall and 40 feet long. A cursory glance reveals that the partition is comprised of plywood and simple 2 by 4 frame. One side has been covered with what appears to be covered by several layers of paper. More interesting than the wall's construction, however, is what's being done to it. All along the length people are writing and drawing on the surface of the partition. Some work together in groups, all contributing to a single composition while others stand to the alone working on their own projects. Still more hang back and merely watch what others are creating. The subject matter is eclectic at best, as is the range of skills of the artists. Some people are not even drawing. Some merely write messages or words; pop culture references and slang.

Following the example of other newcomers you proceed to the side of the wall where there is what can only be described as an overhauled deli number dispenser. People are tearing pieces of paper off of this and applying them to the wall, creating new blank space on which to work. You grab a small piece and procure a colored pencil from a woman who has just leaving. Now that the moment has come to create you feel a momentary writers block. You begin by writing your name and entertain yourself doodling random circles while you wait for inspiration to hit. But, before you can come up with anything worthwhile, an alarm sounds and a message plays instructing you to back away from the wall.

As you move away, previously overlooked machines hidden in the corners of the wall spring to life. One , the smaller of the two, moves quickly to the center of the page and begins circling erratically as music is played out of its speaker. Meanwhile the larger machine begins to systematically moving about the wall. Starting on the left, it travels from the top of the wall all the way to the bottom. Everywhere that the robot travels, the white paper becomes completely black. As the smaller robot moves over the newly blackened paper, the music changes subtly, becoming sadder and slower. People stand and walk along the path listening to the unique composition. Eventually the entirety of the board has been transformed; erased of the unique art that once adorned and now completely black. Their jobs complete for the moment, the machines retreat back to their hiding places and users are once again invited to the board as fresh paper is dispensed. The cycle begins again.

tech overview

breakdown of physical/digital components

hecktor clone for writing

vertical plotter for eraser

heat source

heat gun

hair dryer

heat lamp

software overview

processing [editing, image compilation, generating note info, communicating with hardware]

gsvideo library [image grabbing]

osc [inter-program communication]

audio creation software [puredata for .wav/.mp3 sample playing/cutting, jeskola buzz for software synthesizer audio generation]

The technology that is used in our project varies from physical components to programming languages and software. On the physical side our project is broken up into three components, the hecktor clone for reading the data that people put up on the wall, the vertical plotter that erases what people have put up on the board and the computer that crunches all the input data and then outputs it as sound. The first component the hecktor clone is a simple off the shelf webcam attached to two bi-polar stepper motors controlled by an arduino microcontroller powered by a atmega168 ic, which is fed commands from the computer. The stepper motors control the position of the webcam. Meanwhile the webcam reads what is on the board and sends its information to the computer. The second component is the vertical plotter that erases the data people have put up. It is just a heat gun from a hair dryer attached to two bi-polar stepper motors that control its x and y position. The heat changes the thermal paper that's on the board black erasing what was there. The motors on the vertical plotter at run through another arduino that controls the logistics of its movements.

On the software side the arduino uses its own environment that is downloadable off of the internet at http://arduino.cc/. It is a java like environment based off of the programming language processing. There is a lot of support and work being done on the internet with this particular microcontroller so it made it an ideal choice. The computer that where using to take all of the input data and to output sound using Windows XP as an operating system. It is using the programming language processing to take in the information from the webcam in the form of a picture. From that it calculated pixel color, luminance data and also basic shape recognition and then this is passed off to a program and it interprets the data and outputs sound based on the above mentioned data. In order to grab the data from the webcam processing it utilizing a library called gsvideo and it makes it very easy to get data directly from the webcam. Now there will be other programs that will interpret the data that processing will get so we use osc for inter-program communication and it makes it simple to pass data from one program to another as well as to different computers and operating systems. The programs that will make the music are written in programming language called puredata and also jeskola buzz a program for synthesizer audio generation.

problems

software complications, fire, bears, motor control, communication issues

plan of creation

list stages of production/development

research stuff to buy [torque for stepper motors, usb controller, heat source for exposing paper]

purchase!

build/program

refine

roll out!

6.4 Name Idea's For the Project

Name Idea's for the Board!!

The Pixels of ...

The Board of ...

10 Music in a Post-It Note

Sticky notes

sticky song

sticky jam

(something with sticky?!)

stickit playit wipeit

sticky player

sticky pixels:

an office supply serenade

pixel player

pixel song

pixel music machine

singing pixels -> I like this- seth

the pixel organ

8 pixelbotsynth

pixelsynthbot

stickypixelbot

(i could do this all day - JR)

messages for music

pixelBlitz

readSonic

The Sticky Pixel Analysis Music Machine

Fully Automatic Musical Square Analyzer - FAMSA

Fully Automated Musical Square Analysis Machine - FAMSAM

Sticky Pixel Analysis Machine - Spance

The Post It Cycle - PIC

New Age Music Synthesizer - NAMS

New Age Sound Synthesizer - NASS

fully automated musical office supply synthesizer - famoss

Sticky Pixels : Office Supply Serenade

The Sticky Pixel Analysis Music Machine - SPAMM

Fully Automatic Musical Square Analyzer - FAMSA