

# **Real Time Tactile Image Generator for the Blind and Visually Impaired**

**Category: Computational System and Analysis  
Project ID: J0810**

## **Table of Contents**

Abstract.....	3
Engineering Goal and Purpose.....	4
Background Research.....	5, 6, 7, 8
Design Criteria, Constraints and Test Plan.....	9
Preliminary and Final Designs.....	10,11,12
Prototype.....	13
Initial Testing and Evaluation.....	14
Redesign, Testing, and Evaluation.....	15, 16
Design Results.....	17
Conclusion.....	18
Acknowledgements.....	19
Bibliography.....	20

# A Real Time Tactile Image Generator for the Blind and Visually Impaired Abstract

#J0810



253 million people on this Earth have visual impairments. I feel fortunate to have none of these impairments, but I wanted to do something to give these people some imitation of sight, and thus the birth of my science fair project. In this engineering project, I have made a mechanism to convert an image from a camera into a tangible surface that can be felt by the user. A tangible image is an image that can be perceived by touch. A classic example is pin art, like the one in the image below. If I could successfully create one of these using live feed from a camera, the visually impaired could use it in their daily lives. In order to create my mechanism, I took an image from a camera. Then I pixelated the image to the number of pins used to portray the image. I used a 6x6 array to prove the concept. By taking the average luminosity value (brightness) of each sector, I could drive the height of the pin to represent the average brightness. I used a camera which fed an image into OpenCV (an open source computer vision program). The program pixelated the image and sent the brightness value of each pixel to an Arduino which drove each of the 36 rods via servos to create a tactile map of the image brightness. In the future, I would like to use smaller servos to make my design more portable. Also instead of using luminosity, I could use a Lidar (laser based radar) which can determine the distance to an object to create a true topographical map similar to a pin art toy.

## Engineering Goal and Purpose

### Engineering Goal:

The goal of this project was to successfully create tactile image by taking a digital image, breaking it up into a 6 by 6 array, and then assigning each array to one push rod which will go up and down based on how far away the object is. There by allowing a blind person to "feel" an image of his surroundings in real-time.



Purpose: The purpose of this project was to help blind and visual impaired people "see" their surroundings by creating a tactile representation of their surroundings. With the help of this technology, the blind and visually impaired navigate more efficiently through the world. The original thought was to create a small disk that could be worn in the palm of the hand which when pointed at an object would create a pressure or contour in the palm proportional to distance. However, because we are using off-the-shelf parts the prototype is much bigger than I expected. The next step is to work towards making this so small that it fit in one's palm and be used for daily purposes.

## **Background Research**

What is a tactile image? Applications?

According to madilexicon.com, a tactile image is an image of an object as perceived by the sense of touch. This means that instead of using our eyes and how light is reflected to perceive color or an image, we are using the depth and feel of a surface to determine the surface's appearance. Some applications of tactile images are explored in this project. A tactile image can be used to help the blind in their everyday life detect how far away something is from their location. Pictures can also be shown using tactile images with more sophisticated technology.

### **What technology will we use?**

OVERVIEW- I will use a camera to get the image, then the image will be merged into a smaller number of pixels by python code which will then send the "image to servos via an Arduino communication with servos via I2Ccommunication protocol, to move which will move pushrods to create a tactile representation of the image the camera sees

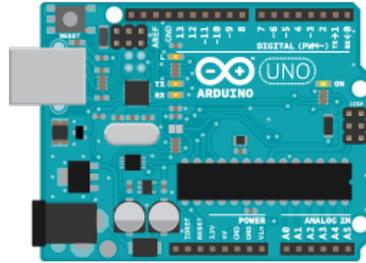
Python Code:

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales.



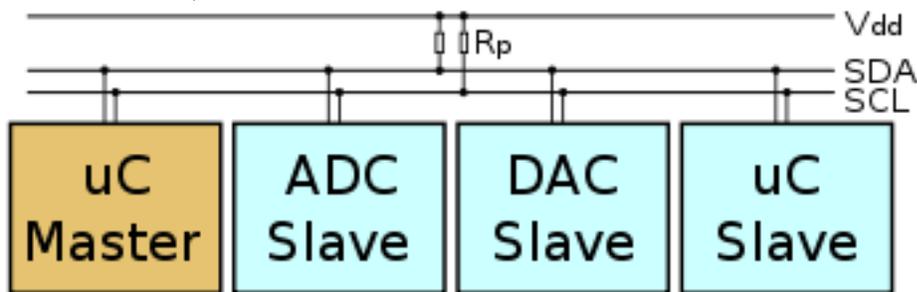
Arduino:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. We used this to communicate with the I2C and the Python Code. (Source: arduino.cc)



I2C:

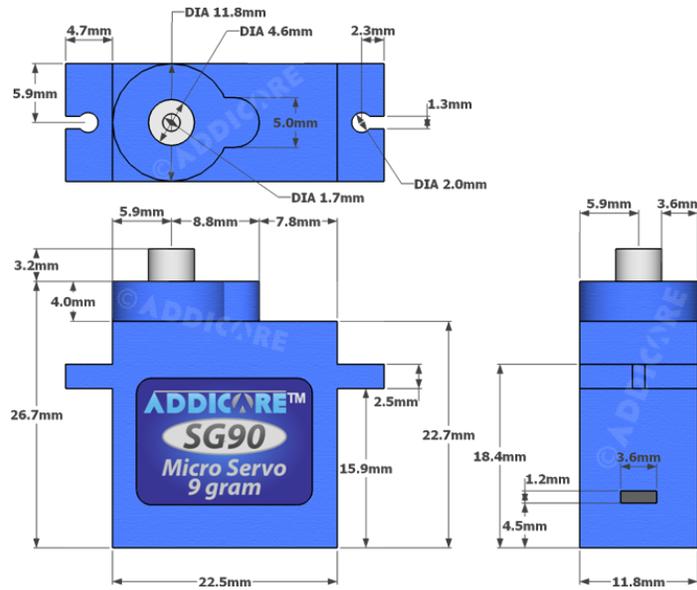
I2C is a serial protocol for two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems. It was invented by Philips and now it is used by almost all major IC manufacturers. Each I2C slave device needs an address – they must still be obtained from NXP (formerly Philips semiconductors).



I2C bus is popular because it is simple to use, there can be more than one master, only upper bus speed is defined and only two wires with pull-up resistors are needed to connect almost unlimited number of I2C devices. I2C can use even slower microcontrollers with general-purpose I/O pins since they only need to generate correct Start and Stop conditions in addition to functions for reading and writing a byte. Each slave device has a unique address. Transfer from and to master device is serial and it is split into 8-bit packets. All these simple requirements make it very simple to implement I2C interface even with cheap microcontrollers that have no special I2C hardware controller. You only need 2 free I/O pins and few simple i2C routines to send and receive commands. The initial I2C specifications defined maximum clock frequency of 100 kHz. This was later increased to 400 kHz as Fast mode. There is also a High speed mode which can go up to 3.4 MHz and there is also a 5 MHz ultra-fast mode. We used the I2c to drive our servos. (Source: i2c.info)

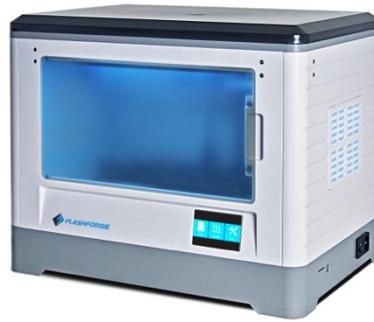
### SG90 Servo:

Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware. We used these servos to drive the push rods up and down which could be felt by the person.



3D Printer (Flashforge Dreamer):

3D printing, also known as additive manufacturing, refers to processes used to create a three-dimensional object in which layers of material are formed under computer control to create an object. We used the Flashforge Dreamer printer in specific to print out our materials. (Source: Google)



## **Design Criteria, Constraints, and Test Plan**

### Design Criteria:

Our design has to be able to take an image from a camera and make the image tactile for visually impaired. For ease of execution and proof of concept, we will use a 6x6 array of pushrods. We will apply an algorithm to the image to pixelate the image into a 6 by 6 array. The average luminosity of each pixel will be used to drive a piston up and down by an amount proportional to the brightness.

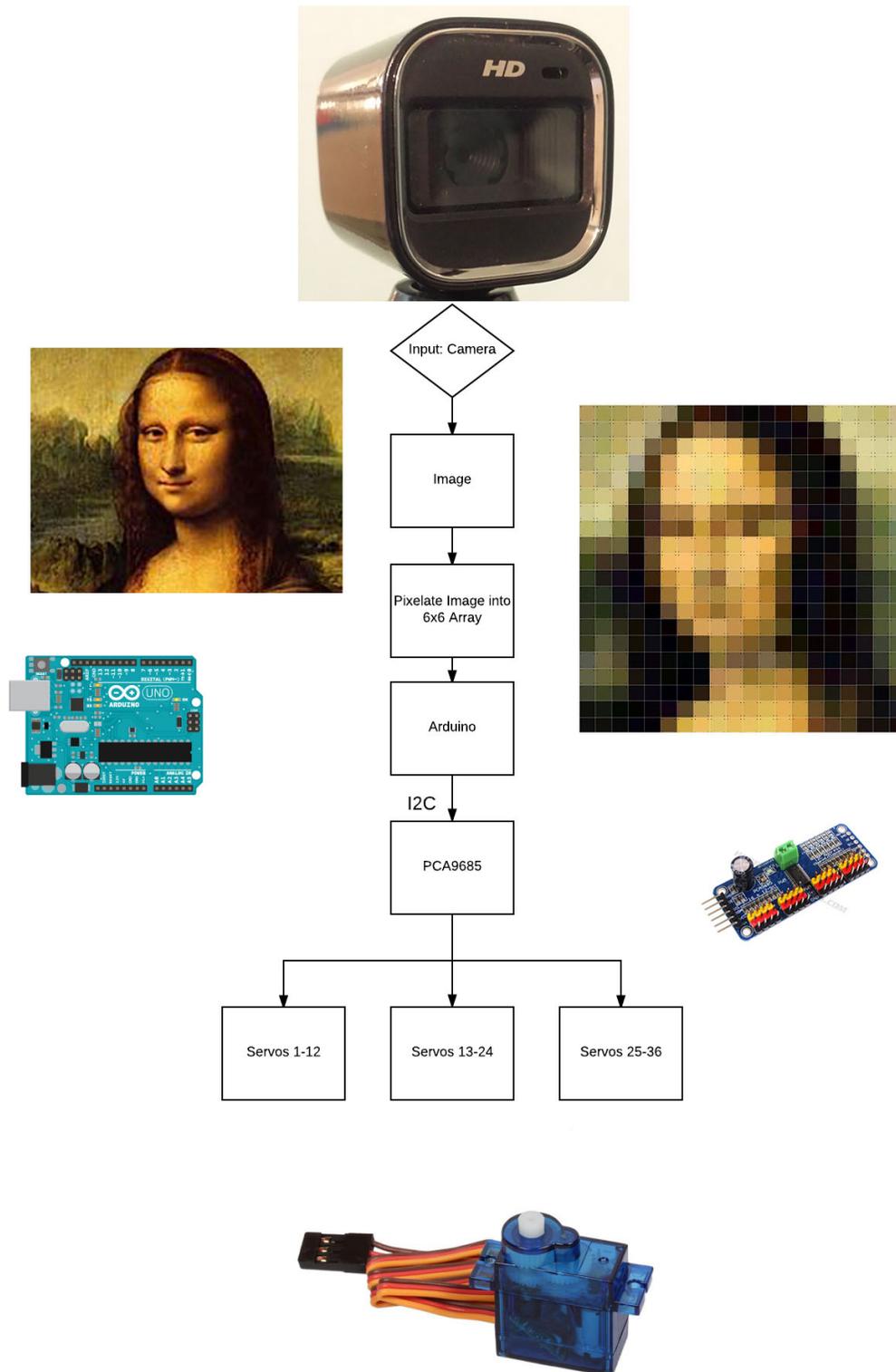
### Constraints:

- Time - The project needed to be completed in 6 months prior to the beginning of the Alameda County Science Fair
- Cost - The design had to be of a cost affordable to the consumer in order for it to have maximum benefit amongst people.
- Size - A blind person cannot be walking around with a huge mechanism to help them navigate through their lives. The mechanism must be small and portable.
- Accessibility to materials (Carbon Fiber vs. Plastic) - Ideally, I wish the mechanism of the device could have been made out of carbon fibers which are both light and strong. However since my 3D printer only prints plastics such as ABS, I made the device in ABS.
- Weight - Just like size, a blind person cannot have to carry such a heavy mechanism. The mechanism should be of weight that will not tire the wearer, for it will be used in everyday life
- Safe - The mechanisms cannot have sharp edges, for they prove a safety hazard.
- Responsive (< 0.5seconds) - There cannot be lag in the computing time, for a blind person may get the signal too late

### Test Plan:

I will attempt to construct a fixture that uses motors to push 3D printed rods up and down. The user will be able to feel the rods, and with the use of a camera, can feel an image based of luminosity values. These values will be acquired by making an image a 6 by 6 array and taking the average brightness for every section. The rods will be pushed out an amount proportional to the brightness.

## The Process / Flowchart:



## Preliminary and Final Designs

My project design varied much during this project. The original idea was to make the design fit in the palm of your hand. I bought very small motors which had its benefits because people could carry this with them. However, there were issues. For example, I had to convert the motor motion from rotary to linear. Rotary Motion is when shaft in the motor spinning in a circle. Linear Motion is the up and down motion. All the motor designs had rotary motion, but the SG90 servo, my final design, was the easiest to convert to linear motion. For the time given, my original design wouldn't be possible so I had to find an easier option.



After looking at the flaws in my first design, I turned to use larger motors that were easier to operate. The issue was that the motors didn't have a position control. If I applied power to the motors, they spun continuously and I wasn't aware of how far up a nut could go up the rod.



After extensive search for the right motor, I found the SG90 servo and it was a good compromise. The SG90 had position control. I could set exactly where I needed the servo to go and there was no continuous spinning. I could stop the motor at exact spots, which I could not do with the first 2 designs. In addition, these servos were easy to convert from rotary to linear motion. The servos came with arms, and with the new fixture designed, it worked perfectly. The only issue was the size, and the original idea of having it small couldn't be possible for now. However, as a follow-up project, I will try to make the first design work so that

the visually impaired can use this in their everyday life.



The prototype was designed to receive a visual image from a camera and create a contour map of the image that can be felt by a blind person. The original idea was to create a design similar to a pin art sculpture as shown below.



Each pin would be connected to a motor based actuator that can push the pin to a height that is set by the brightness of the image. Ideally the height should be set by the distance which would be measured using a Lidar similar to that used in self driving cars. However, for the purpose of this project, we will use a camera and display light and dark on the pins rather than distance from a Lidar.

For the prototype to show proof of concept, I will use a 6 by 6 array of pins moved by hobby servos. I will use OpenCV, the computer vision library in Python to pixelate a live image from a camera. The brightness of each pixel will then drive the height of the servo to a distance proportional to the brightness.

The Arduino connects to a circuit board that generate signals that drive the servos up and down.

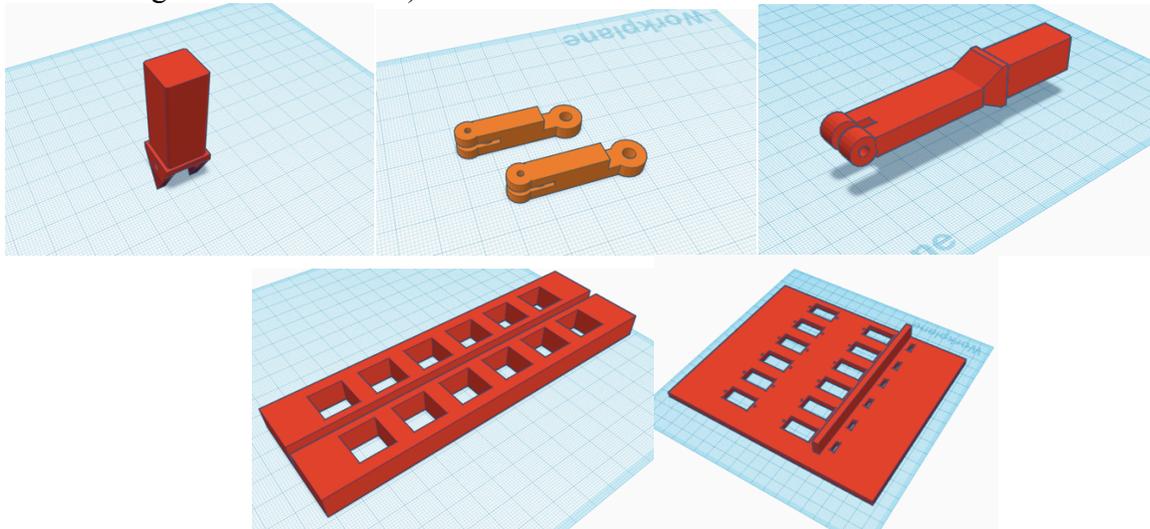
# Prototype

## Materials:

- SG90 Hobby Servos (x36)
- 3D printer
- Computer with Python and Arduino programming software
- Spring Pins
- Arduino Microcontroller (Uno was used in this experiment)
- PCA9685 servo driver
- Jumper Wires
- Camera

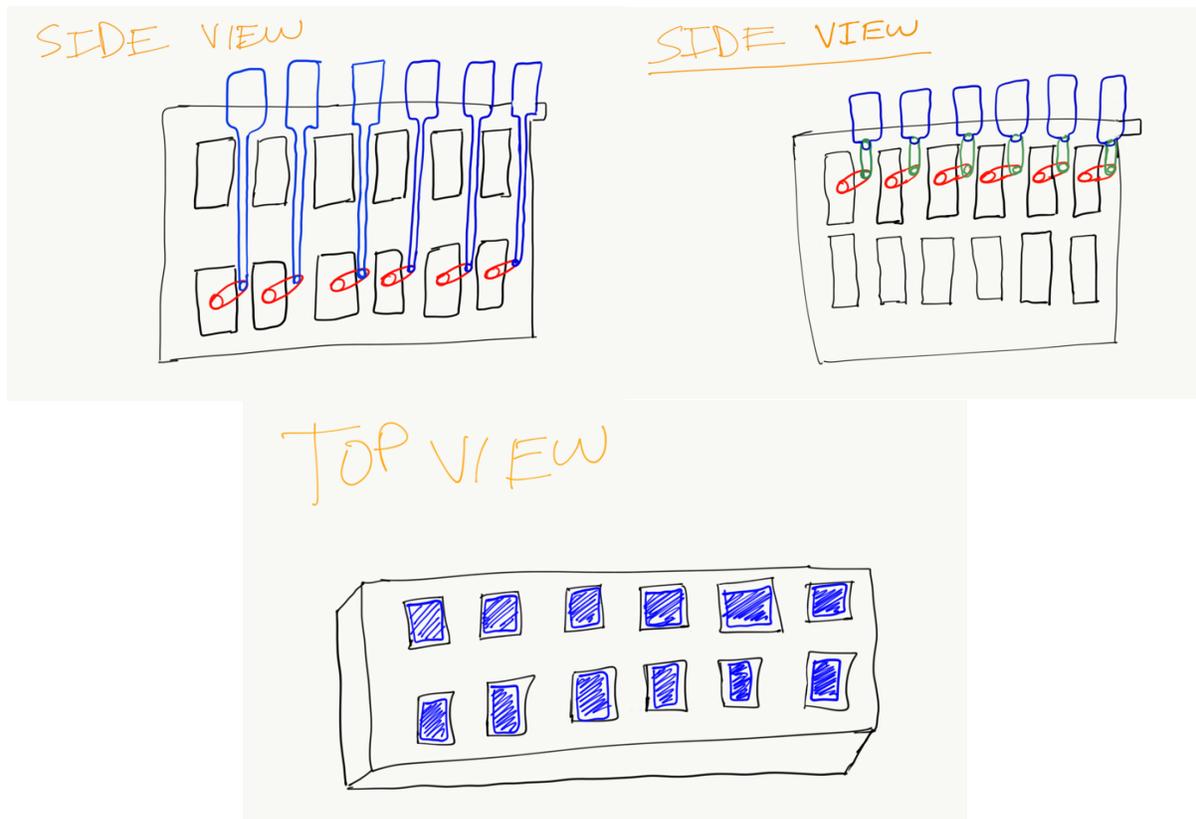
## Procedure:

1. Design the fixture to hold the servos, the push rods, and pistons (Some images of the design are shown below)



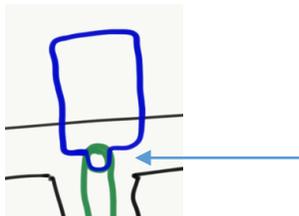
2. 3D print the designs using the 3D printer
3. Install the servos and push rods so the fixture works
4. Take an image from a camera. Write python code to break the image up into a 6 by 6 array. Depending on the luminosity value of each sector, make the pistons go up and down using Arduino code. The Arduino should communicate with the PCA via I2C to drive the servos.
5. After working with a still image, write python code to take live video to pixelate the video into a 6 by 6 array that can continuously update depending on the video.

## Initial Testing and Evaluation

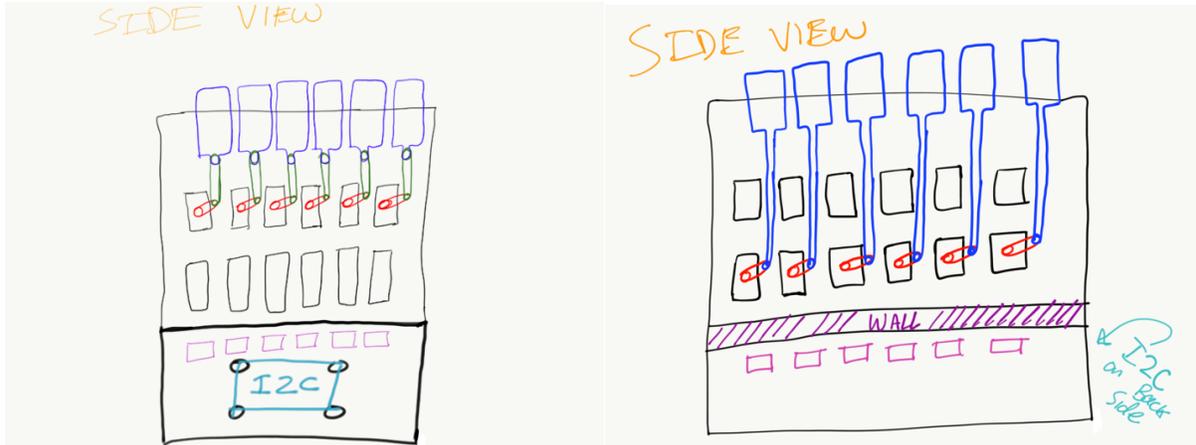


Above are some designs for the motor mounting fixture. At first this idea seemed to have no issues, but under further inspection, some problems arose. First, there was no mounting space for the I2C. Second, when the motors were in the down position, the arms would break because of the lack of room and proximity of the next motor. I had to fix this problem, for the arrangement of the arms should not hit in all situations. Third, if one arm went to the right, and the other went to the left, they would hit and I had to make the fixture so that there were no issues in all situations that the unpredictable software may go. I also needed a way to mount 3 fixtures to make a 6 by 6 array, which wasn't included in this design.

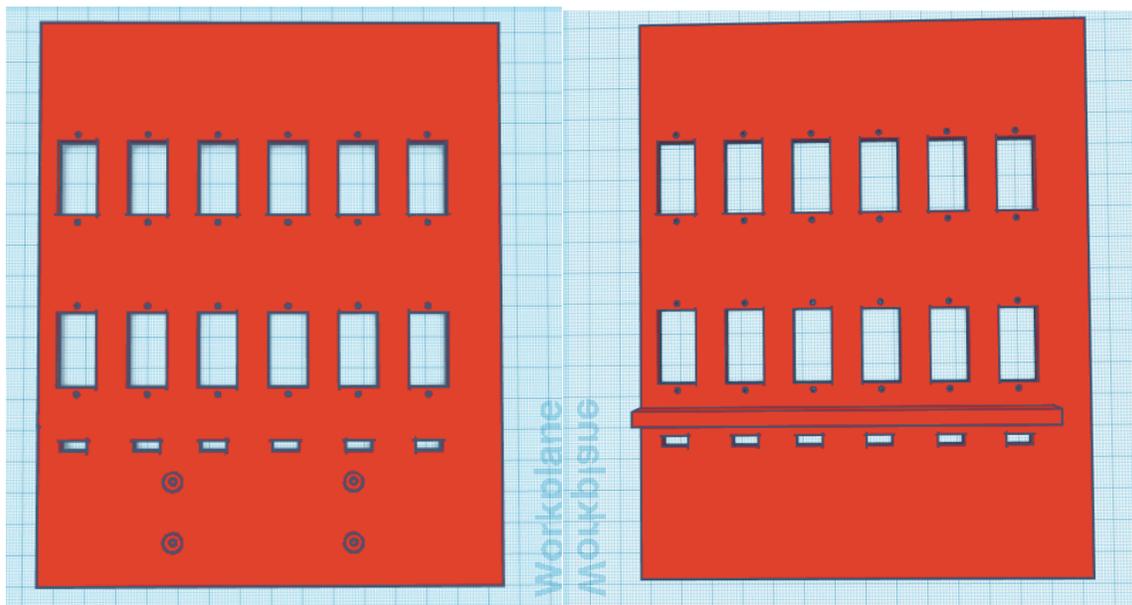
To connect the servo arms to the push rods, I had to 3D print several plastic pins. These pins sometimes were shabby and were of low quality. The pins also snapped sometimes and got jammed in the rods. To get the pins to go in, I had to drill out holes in the push rods that were enough diameter. The plastic being feeble, usually broke which caused a delay in the project.

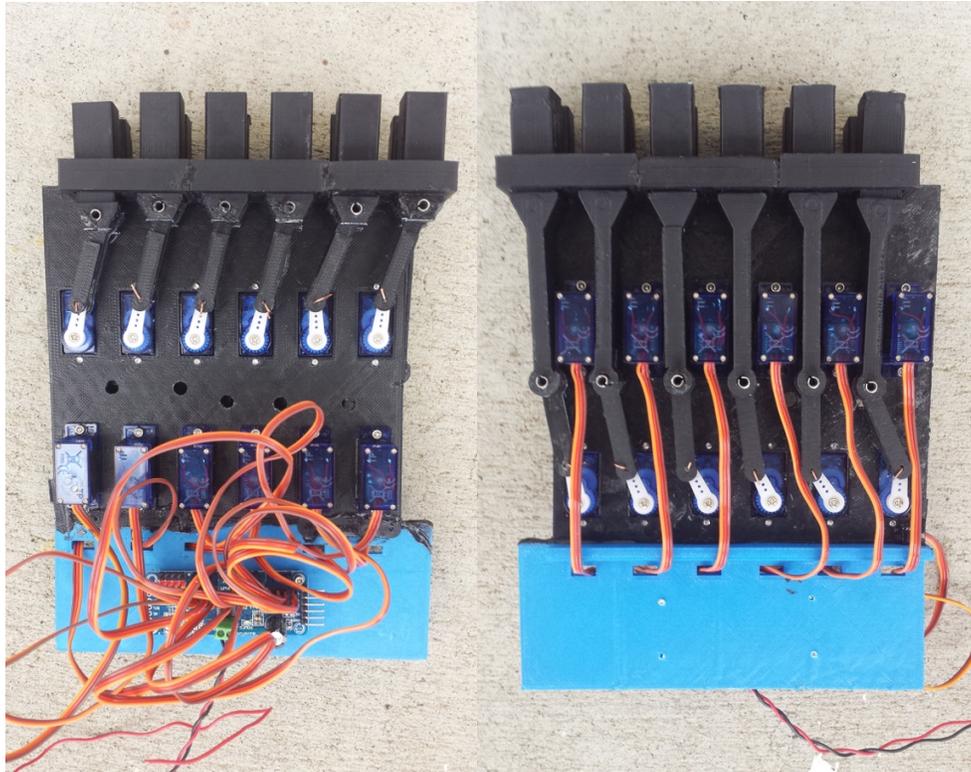


## Redesign, Testing, and Evaluation



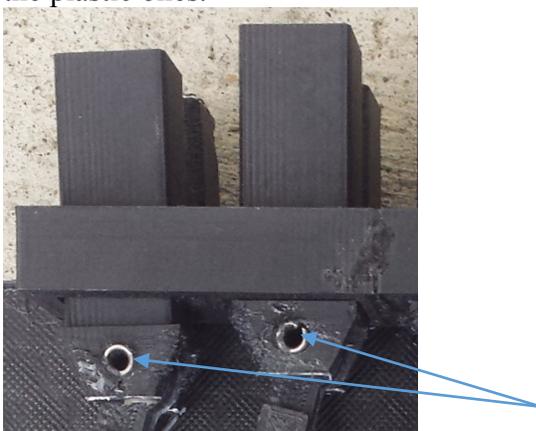
I was able to solve these problems in the second design. I made an extension that could go at the bottom of the fixture, which contained bosses that could mount the I2C. This extension also contained a wall which could be used to better organize the wires. The distance between the motor mounts was increased to anticipate all possible ways the motors could turn. To complete the design, I mounted 3 of these individual fixtures to make the top surface a 6 by 6 array. Above is the design for how I was going to tackle the challenges. Below is the 3D design that I printed and below that is how the fixture with the motors looks.



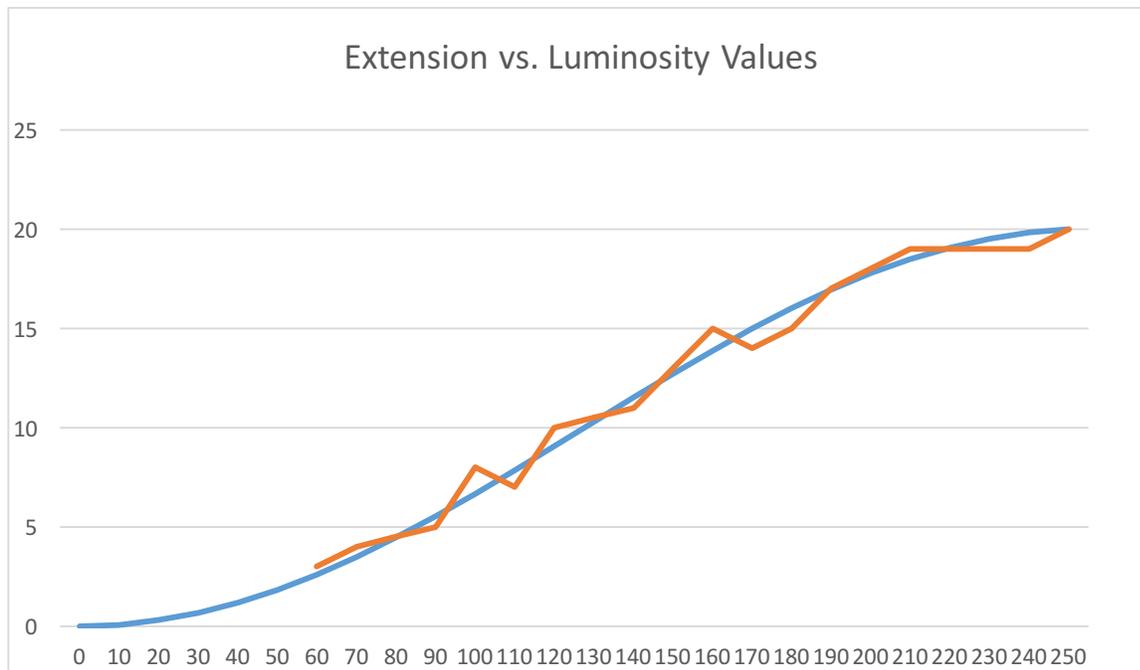


This design was quite successful. The push rods went up and down based on the luminosity of the image and the design had no technical errors. The wall proved useful and helped prevent disorganization amongst the wires. The spacing between the motors was just right and the arms never crashed into each other.

To solve the problem of the pins in the initial testing, I replaced the plastic pins with metal pins. These metal pins were slightly harder to install, but were more strong and smooth compared to the plastic ones.



## Design Results



Above is a chart that shows the extension of the push rod vs the luminosity values. As shown in the chart, the blue line represents the theoretical position the push rod should take, while the orange line shows were the measured extension. I could not measure below 1/2mm accuracy because of the limited travel on the arm. The theoretical equation is as follows:  $10\text{mm} * (1 - \cos(3.14 * \text{Luminosity Value} / 255))$ . Notice the curve is an S shaped for I am converting a rotary motion to a linear motion.

## Conclusion

Overall, this project can help the blind navigate through their every day lives. If I can reduce the size of my project, blind people can use this in their everyday lives.

### Follow-up Project:

While light and dark represent a visual image, the actual contours of the object may make more sense to a blind person feeling the object because light and dark are not a full representation. It may be better to create a true contour map using a lidar. Portable hand held lighdars may be possible by the use of 2d mirrors made by AG Microsystems.

### Two Axis MEMS Chips for 1xN Optical Switch and Other Applications

This line of AGM products signifies a robust and growing area of development for AGM, one that shows much promise for meeting customer roadmaps for more optical functionality. It extends the scope of the core MEMS technology that represents AGM's expertise.

The AGM Two Axis Micromirror (2D MEMS) product was developed to enable 1xN optical switch products, and is successfully deployed in 1x4, 1x8, 1x16, and 1x32 optical switches. These devices can be packaged in TO46 or TO39 headers, or ceramic surface mount packages allowing for smaller size and lower production costs. The AGM MEMS chip offers superior optical performance. The mirror angle is very stable and repeatable over time, and offers low susceptibility to shock, vibration, and temperature changes.

The AGM Two Axis Micromirror technology has also been successfully employed for applications related to Tunable Filter and Optical Channel Monitor (OCM). In addition, AGM is exploring applications of this technology in the emerging areas of 3D imaging, LIDAR and free space optical communication. An evaluation kit offered by AGM, including the 2D MEMS chip along with PCB driver board enables customers to explore the potential applications of this novel technology.

Product specifications for the 2D MEMS 1xN Chip are available in pdf format at the link below.

[Spec Sheet – AGM MEMS Chip for 1 x N Optical Switches and other Applications](#)



## **Acknowledgements**

I would like to thank my science teacher for reviewing my submissions. I would also like to thank my parents for guiding me during this project and my cousin for helping me solve problems I had with the code.

## **Bibliography**

- How to Make Tactile Pictures Understandable to the Blind Reader, [www.dinf.ne.jp/doc/english/Us\\_Eu/conf/z19/z19001/z1900116.html](http://www.dinf.ne.jp/doc/english/Us_Eu/conf/z19/z19001/z1900116.html).
- “Optical MEMS | Micromirrors | Fiber Optic Components | 3D Imaging, LIDAR, Optical Sensors.” AG Microsystems, [agmicrosystems.com/](http://agmicrosystems.com/).
- “I2C - What's That?” I2C Bus, [www.i2c-bus.org/](http://www.i2c-bus.org/).