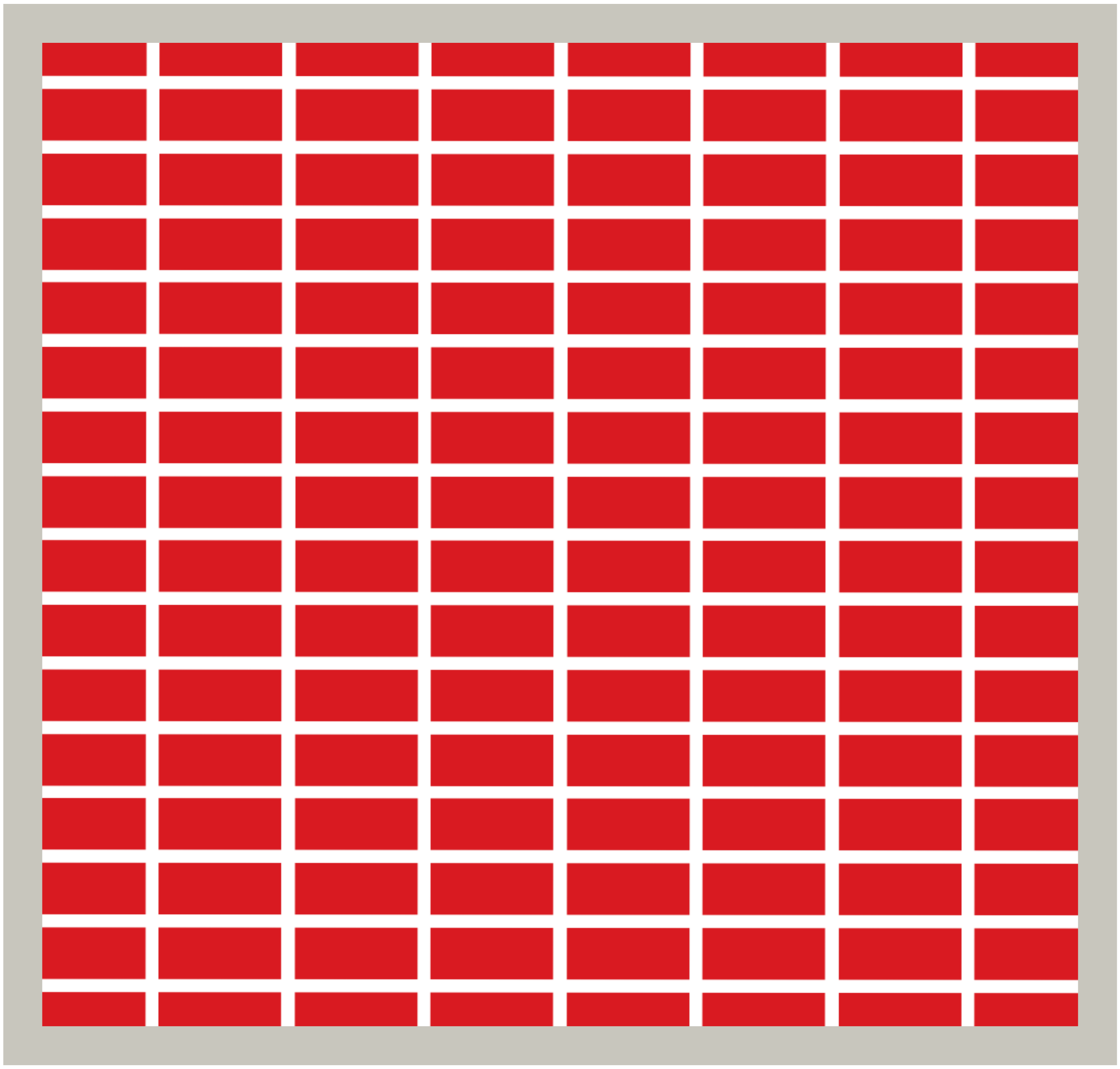


Frames:
A 3-D Image Recording and Display System

Group: 7

Daniel Hureira [CpE], Christian Vartanian [EE], Jamesley Rock [CPE]

Table of Contents

1. Executive Summary:

Visual display systems have captured the attention of many since the first television was created in 1927. With technology rapidly advancing the same has happened to visual display. Throughout history visual projection or as some say projection mapping has changed rapidly. To modern day where projection and even holograms are common. This allows for visual equipment to be cheaper and even more environmentally friendly due to the technological advances. Our aim is to build a lightweight, relatively cheap, display system that can capture and display ultrasonic texture maps of a space or object. Which can be utilized for entertainment and even special forces purposes. By arranging several hundred LEDs in a cube, we will be able to display a 3D image using incoming data from a larger ultrasonic sensor array. Our motivation for picking this design was to not only challenge the skills we have learned throughout studying our undergrad but also to be able to reproduce a visual representation of what technology is capable of and displaying this to our fellow classmates and faculty.

While most people are readily familiar with image reconstruction via optical sources, this is far from the only option. Optical reconstructions confer a great way to simulate the human vision as they are able to capture the full spectrum of color, unfortunately they do not excel at depth and structural reconstruction. If we are willing to exchange color for a high level of depth awareness, a sonic approach becomes available. Sonic image reconstruction has two major advantages over optical imaging, low light operationality and fine readings of true depth (as opposed to a simulated "perception" of depth). By sending ultrasonic pulses outward and calculating the round trip travel time a fine reading of the distance from the sensor can be generated. This allows sonic imaging to be able to accurately represent depth perception, albeit at the cost of any form of color. This form of image reconstruction is performed constantly in nature, most infamously by bats. The ultrasonic echolocation implored by bats allows them to obtain an understanding of their surroundings in low light situations that lets them navigate caves while flying, and thus is of high precision and capable of generating a high fidelity reconstruction of the surroundings.

While adept at generating depth information, sonic sensors are unable to form a recognizable image on their own. One sensor only represents one voxels worth of information and without other sensors to put their own position into context, they are only really able to convey a simple Z-Axis measurement. In order to properly generate a real time 3-D construction, Frames needs to be designed in a manner that allows it to contextualize the information in front of this, the obvious choice to remedy this problem is to simply use multiple sensors. While one could use a single sensor and mechanically move it around, generating a real time image of the surroundings would require one to rotate the sensor at a high enough rate to achieve a fluid motion of image. Film, for example operates at 24FPS which at 1 frame per rotation would require a single sensor to be moved at 1,440 RPM. This is obviously quite fast to be rotating and then subsequently processing image data at. A cleaner solution arises when one considers simply using multiple sensors to contextualize the data on a Cartesian X-Y grid. Forming a clear input that can be processed easily in real time and that has a high enough image density to properly intake the full shape of the forms in front of it.

3-D image reconstruction is obviously bottlenecked by the display system used. Rarely are displays capable of legitimately recreating three dimensional images as they are themselves two dimensional. As such Frames requires a voxel based 3-Dimensional display to be able to properly complete the imaging process. A cube of XYZ addressable LEDs must be deployed in order to properly construct the image that was taken in at the inputs. In designing this cube we will be able to properly show the totality of the incoming data stream and properly display depth as opposed to a simply simulating depth *perception*.

Real Time processing and outputting of this incoming data stream is obviously processor intensive and requires a large amount of I/O ports, as such we have opted to use a powerful and expandable microcomputer to easily handle the imaging process. Specifically, we have gone with the BeagleBone Black for its strong GPIO selection and plentiful RAM.

2. Project Description:

The system is made of three core components, a two-dimensional array of ultrasonic sensors (henceforth known as the "Frame"), a microcomputer, and a volumetric array of LEDs ("Cube"). The *Frame* acts as the input of the system. As Ultrasonic Rangefinder (HC-SR04) units are able to generate a continuous stream of depth data, arranging these sensors in a matrix will allow for a projection map of any space to be captured. This texture map of the space can be directly serialized, passed over to a micro controller unit, compressed via a computer vision program, and then sent over a serial connection (potentially WIFI) to an LED Cube that will then utilize the X,Y,Z data generated by the Frame array to display a real time 3-D representation of whatever is in front of the sensor array.

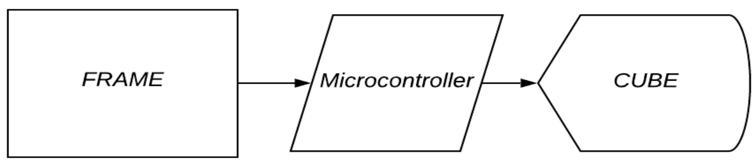


Figure 1: Layout of Project

After being captured by the Frame, it's the job of the middleware running on the microcontroller to quantize the incoming distance readings into discrete levels that can then be displayed by the Cube. The X and Y components of the output have already been discretized as they are the literal X and Y positions of each sensor on the matrix; this leaves the only remaining data that needs to be quantized to be the Z axis. This would be a relatively simple process if only being implemented on a single row of lights, but upon considering the scale of even a single Frames worth of data and its respective LED requirements (a 16x16 Frame would require 4,096 LEDs) it becomes obvious that a form of edge detection and compression is crucial to limiting the build out from becoming a mess of multiplexers and an overall wiring nightmare.

The *Cube* is a volumetric array of LEDs that are able to take serialized and quantized X,Y,Z data and build a real time image map of said data. Each LED will be connected to each other in a way that allows each row to be selected and then specific LED's within that row. In its current design each Z axis row of the Cube corresponds exactly to an individual sensor on the Frame. This layout, while potentially complex to scale, allows the system to have a 1:1 representation of any objects in front of the Frame. As our incoming data is serialized, we will be at the whim of frame rate issues and will need to guarantee a high enough frame rate in order to fluidly display the image. In lay terms it's a three-dimensional display that displays the world the way bats see it, using echolocation.

Certain core functions are still being discussed, as should be expected from any project in its early stages. While the general I/O pathway is clear, the actual mechanisms for handling the data is not. We are currently investigating computer vision programs to better understand image compression and edge detection, as the more digital compression is able to be performed the more scalable the system as a whole can become. Aside from the middleware running on the microcontroller, the question of modularity and interconnectivity has come up. In theory we should be able to create a building block like system for attaching multiple sets of frames and cubes together to create larger displays and capture devices. By packaging both the Frame and the Cube as modular blocks we should be able to rapidly scale the system to fit the needs of the space it is deployed in.

2.1 Inspiration for Sensor Array:

When our group decided to build a sensor array, our main inspiration came from music festivals. We wanted to do a project that would be able to be scalable to music festivals (and be interesting enough to be picked up at music festivals). We also wanted to stand out from our peers and many of the projects that were already being done/past projects. We originally came up on the idea of just using the sensor array to display a 2D image. However, as time went on, we immediately began interested in the LED cube and decided to incorporate that entire component into our project as well.

2.2 Inspiration for LED Cube:

In deciding to build a 3D rendering display using an LED cube, my group came across a couple of different inspirations. Two of the main ones being Four Tet's light installation and Max Cooper's light installation.

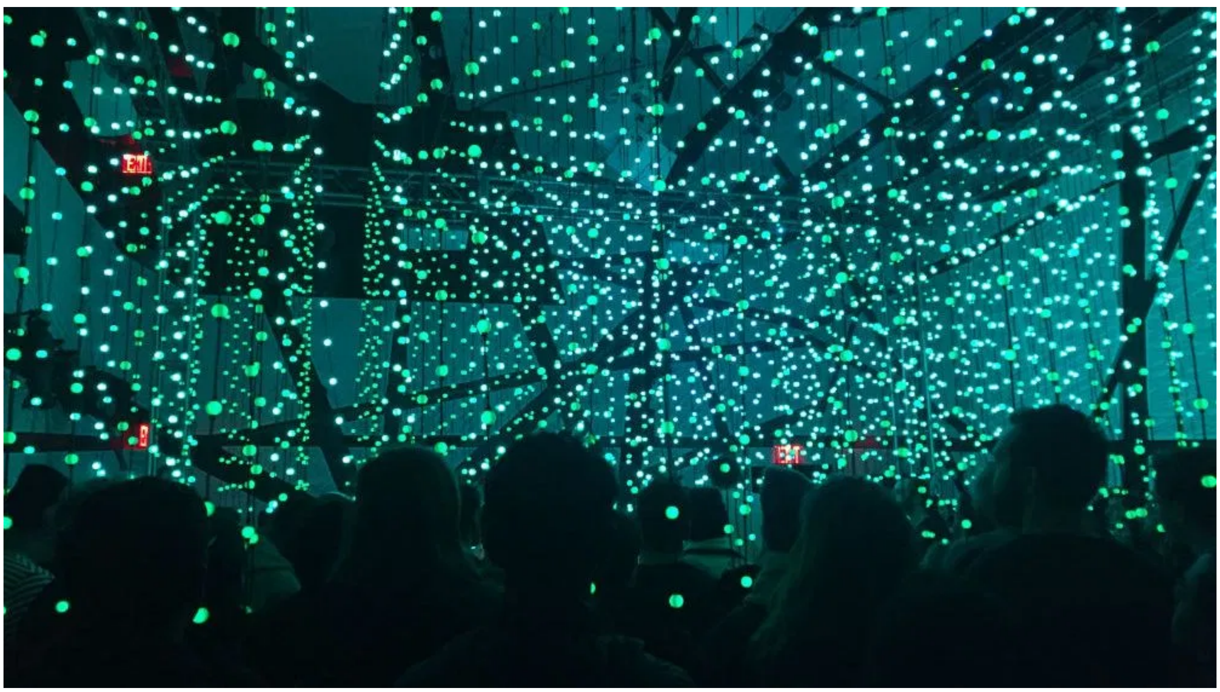


Figure 2: Four Tet Show

Our group was incredibly fascinated with the light shows that these two artists had and decided we wanted to do something involving LEDs/light show. After seeing videos/images of these two shows we instantly realized that implementing something along these lines would not only be a breath-taking display, but it would also incorporate what we've learned throughout our college career excellently. Afterwards we began to conduct extensive research on how we would go about building an LED cube, and how to incorporate it into what we originally had planned for our Senior Design project. We were incredibly fascinated with the fact that our project could be scaled to these venue-sized rooms and able to be much more complex than our system is.

Figure 3: Max Cooper Light Installation

Figure 4: Four Tet Show

Max Cooper and Four Tet's live shows were a large chunk of inspiration for us but we felt that they were missing something. Cooper's show, while stunning, was being displayed on glass beads via a projector, and also couldn't seem to properly display recognizable forms (rather tending to simply rely on the abstraction of form in itself). Four Tet's show, while independently lit, was not displaying forms at all and rather just simple movements, additionally the spacing of his grid was not tight enough for the scale to even be able to recognize any forms. When we saw these installations we wanted to be able to work with the concept of a large cubic arrangement of lights, but with the added benefit of being able to get groups of people to interact or be able to push recognizable forms of our own to the output surface.

Interactivity was a key component of our project. We wanted people to be able to interact with our project as well as witness the spectacle of the lights. This will allow the user to have fun while also learning something interesting.

3. Research into Related Projects:

Our group has conducted an extensive amount of research into projects like ours. We have found projects with LED Cubes, LED walls, or 3D/2D mapping using ultrasonic sensors but never a combination of all those aspects combined. Our project is a culmination of multiple projects in one with more complexity than each. After performing research into 3D Mapping using ultrasonic sound, we came across a video of something similar however it wasn't in real-time and it was being displayed on a monitor instead of on an LED cube. We saw the potential this could have in terms of entertainment and incorporating our knowledge from our college careers into an interesting project. There have been many videos/projects done with ultrasonic 2D mapping, we decided our project could combine the best of both and add more to it. Most of our research found that many of the related projects usually incorporated one or two aspects of the project (whether it be just the LED cube or just the ultrasonic sensors) but never all 3 aspects.

3.1 Examples of Related Projects/Technology:

The following examples demonstrate a few of the related projects we have found during our research. This is from where we drew inspiration on how to do our project After each image/example there is a caption or small paragraph explaining how we drew inspiration from thigs project.

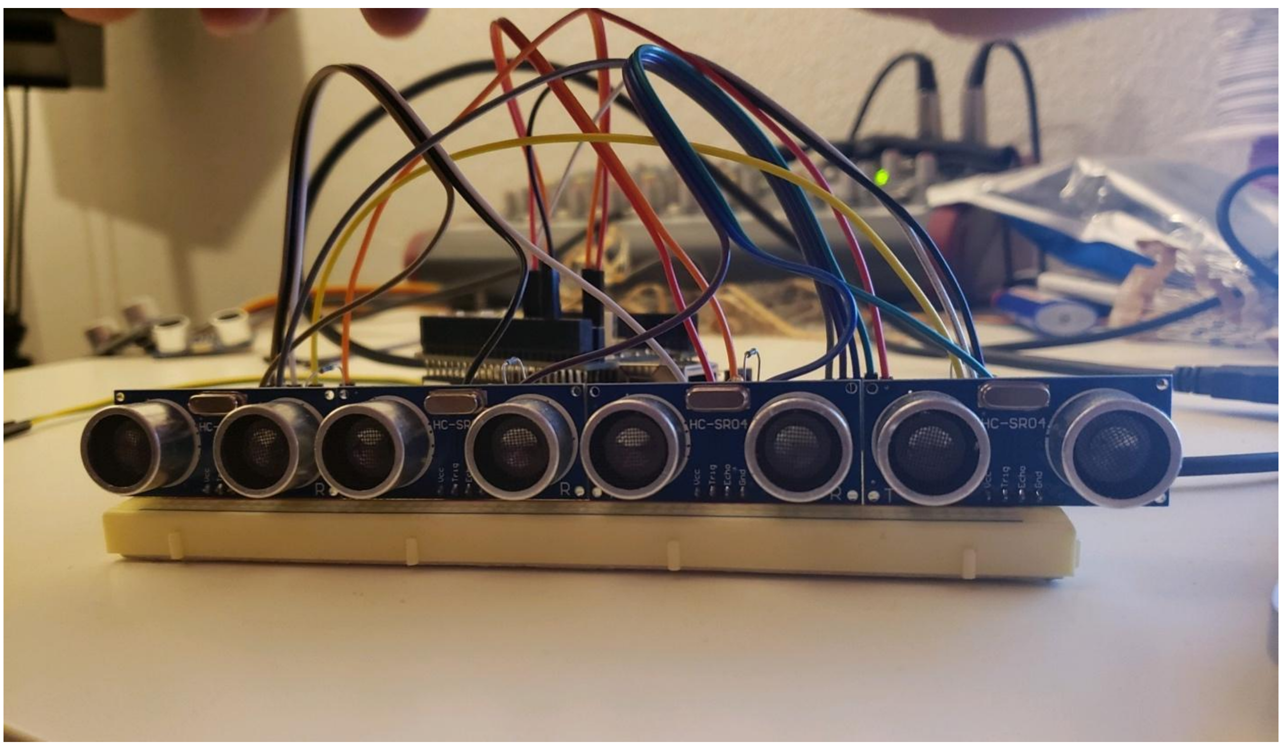


Figure 5: Prototype built of Ultrasonic Sensor Array

Before entering Senior Design, our group member built a prototype of this project as a way to test these ultrasonic sensors he had recently purchased. Our project stems from that prototype while adding more to it.

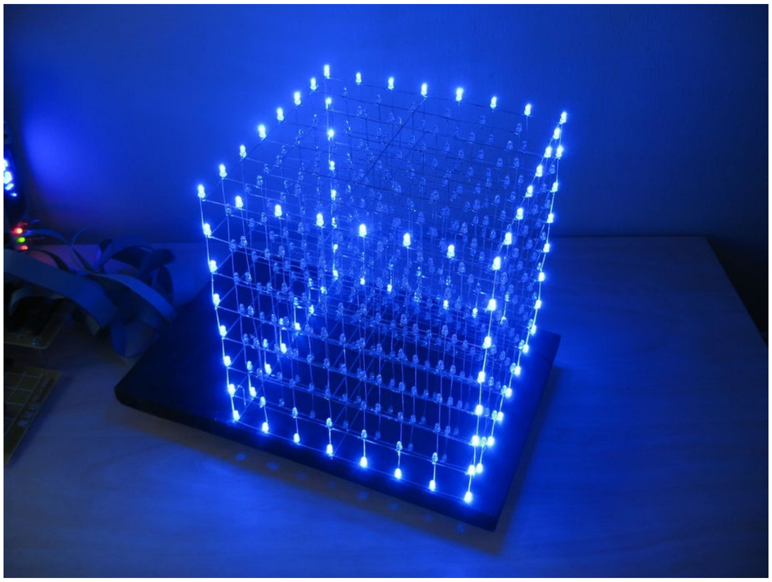


Figure 6: LED Cube (picture printed with permission of Christian Moen)

This LED cube was the main inspiration behind the LED cube portion of the project. However, contrary to our project, this LED cube lacks certain features that we wanted to incorporate. This cube has many preprogrammed effects on it, we wanted to have the LEDs move in real time and respond to input given to the sensor array in real time. Furthermore, we wanted the cube to be larger (the cube shown is 8x8x8) and display input given to the sensor array in 3D. This means, that if the sensor array picks up a hand, the LED cube will display a hand in 3D.

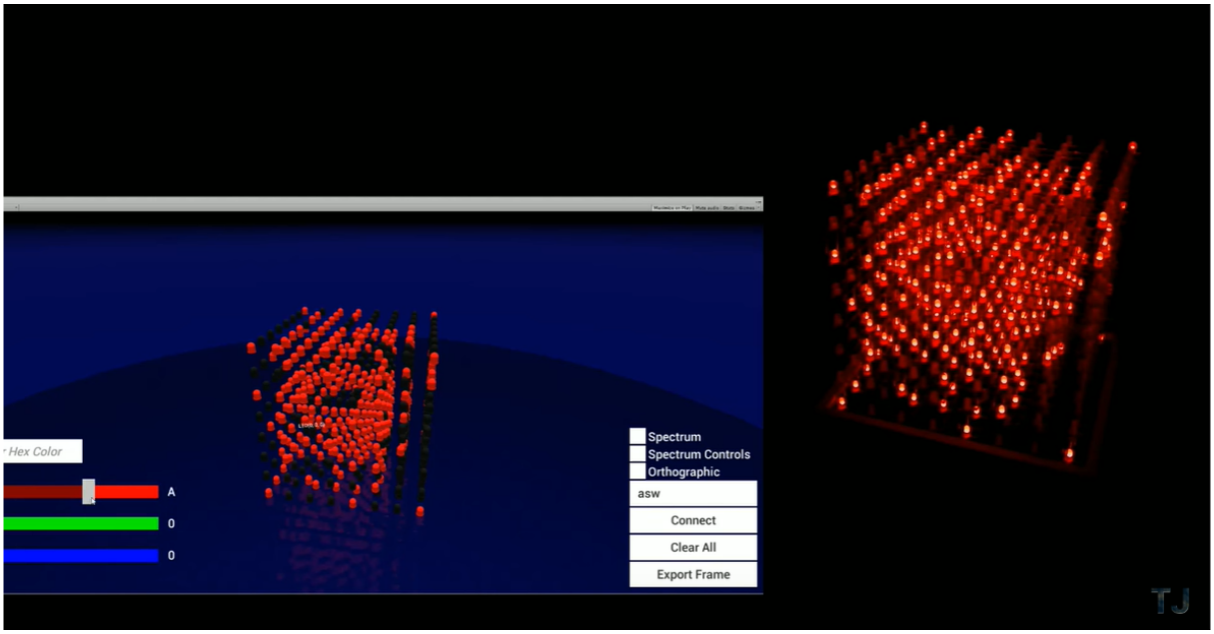


Figure 7: Manipulating LED Cube using Software

The above image is more closely representative of what we want to do with our LED cube. This image comes from a YouTube video of an individual who connected his LED cube to an Arduino board and by using Unity he was able to (in real-time) change and control the LEDs on his cube. The picture on the left represents what he is seeing in Unity, and the image on the right shows the LED cube in real-time responding to his changes in Unity. This is closely resembling what we want to do with our LED Cube however we want it to respond to input given directly to it from the ultrasonic sensor array.

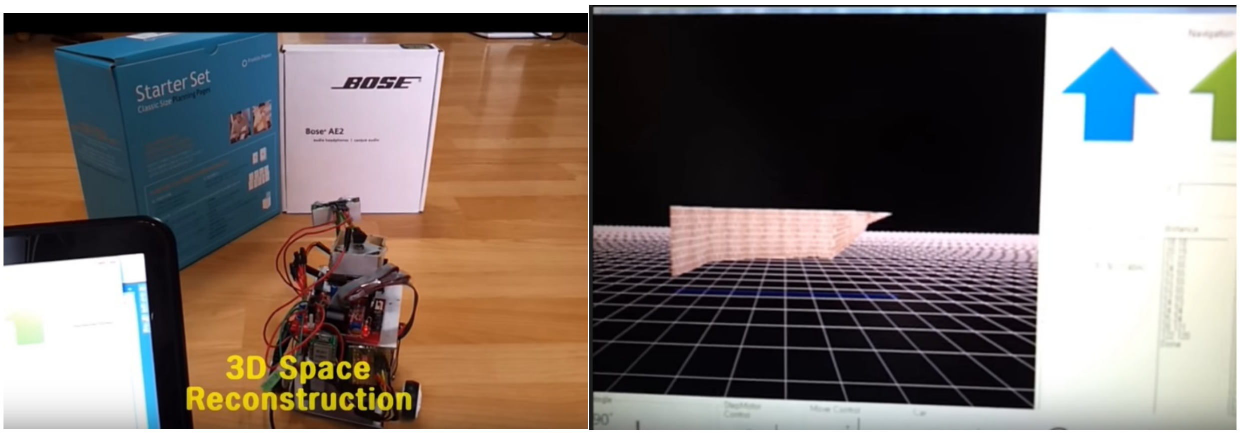


Figure 8: Example of 3D modeling using ultrasonic sensor

The above images most closely represent what we want to do with our ultrasonic sensor array. These images come from a YouTube channel in which the user connected his ultrasonic sensor array to a miniature robot and the robot moved left and right and due to this the ultrasonic sensor captured what was in front of it, then it displayed the image in 3D on the user's laptop. This is closely what we want to do, however we want to scale it and we also want the 3D image to display on an LED cube, and a 2D image on the monitor.

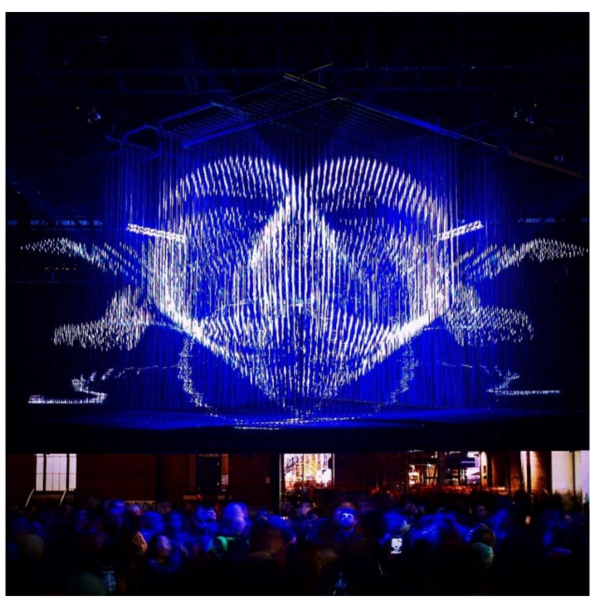


Figure 9: Max Cooper Aether Installation

When doing the initial research on our project, these concepts formed the basis of our project. We wanted to take these ideas and expand on them, improve them in some way and put our own flare on their ideas.

In the following sections, we will expand on why we chose some of the design features we did, as well as many of the many theoretical concepts behind our choices and how we will begin to implement our project.

3.2 Related Design

A design by the name of 2.5D vision utilized many of the concepts and technology we originally thought of using when coming up with our invention. The idea for their experiment was to create a 3D like experience through listening to audio and being able to feel a sensation of being where it was recorded. This was done by taking two audio recordings, and visual frames and turning them into binaural audio and predicted signal. The monoaural audio is converted into binaural audio by leveraging video. This invention is valuable in immersive spatial sound, audiophiles, AR/VR applications, and social video sharers. Similar to our design being a virtual display system. With Binaural audio it provides the listener with a 3D sound sensation, allowing rich perceptual experience of scene. This technology is scarcely available due to requiring expertise and expensive equipment that is also very pricey. This is done by injecting the spatial information contained in video frames with a monaural audio stream. Binaural audio is usually done by using two microphones attached to the ears of a dummy head. The rigs, two microphones, and the spacing play a key role in approximating how humans will receive the sound signals. As a result, when played through headphones giving the listener the 2.5D visual sound and/or 3D sound sensation feel of being where the recording was made. In terms of display system is next level due to not being something that cannot be experienced anywhere.

There are various parts to this experiment, one of them is ITD also known as Interaural Level Difference, due to the human auditory system using two ears to extract sound sources from complex mixtures. Lord Rayleigh theory stated sound locations are determined by time differences between the sounds in reaching each ear. From different studies people perceive the world by combining a number different sensory stream and or senses together, where visual and audio are key. Visual and audio convey huge amounts of spatial information. Spatial information also known as geospatial data and also geographic information is information that identifies geographical locations of feature and boundaries on Earth. This identifies many of Earth features such as natural, constructed features, oceans, and much more. Below is an image which visualizes key region of visual network focuses when performing a mono2binural conversion. This image also gives an indication of what spatial information looks like when pairing image frames with audio. Technology has become so advanced to now see where objects are and how a location is laid out. Also hearing them with sound emitting objects indicates their location, surfaces, material and dimensions. Visual and audio work in convert to interpret spatial signals.

Figure 10: Neural Network

Another Significant part of this experiment is deep convolutional neural network that can decode single channel soundtrack into binaural audio with injecting visual frames and information about object and scene configurations. Being where 2.5D visual sound is a visual steam which would lift the flat single channel audio into spatialized sound. For this experiment the encoder-decoder style network takes a mixed single channel audio and the visual frames that go with it to create a joint audio and video analysis, which will then go on to predict two channel binaural audio agreeing with spatial configurations in the video or frame shown in the figure below. When listening to the predicted binaural audio listeners can feel the locations of sound sources as displayed earlier.

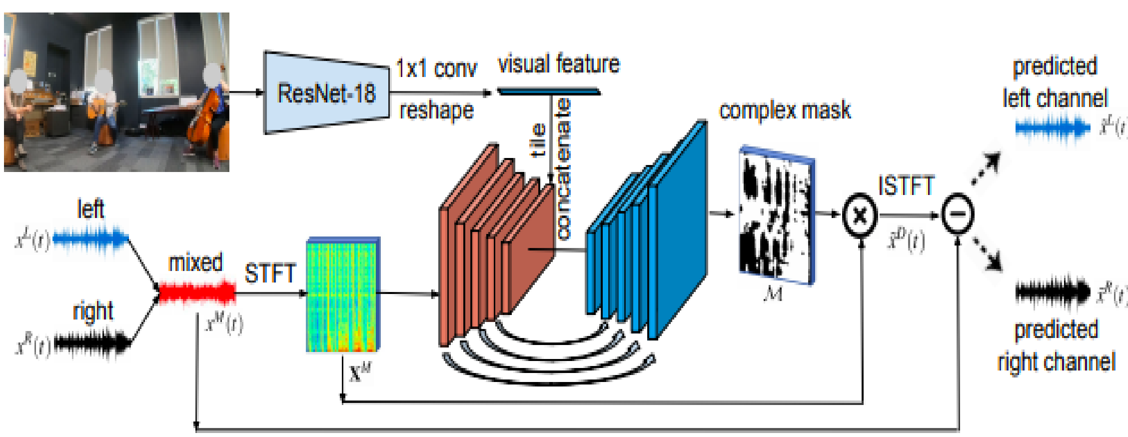


Figure 11: Mono2Binural

In figure shown above what is happening is the mono2binural takes two separate mono recordings and mixes them. The two mixed monaural audios then go to STFT. STFT stands for short time Fourier transform it is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. STFT frames are used in this experiment the same way as it is used in equation 3, by taking the time frame of the mixed audio to predict the frame. Generating an audio spectrogram XM during the prediction. The visual frame using resnet-18 convolution and passes through the brown tile concentrate into the blue concentrate and then to the complex mask. From here the visual frame and mixed audio gets predicted into a two-channel binaural audio output that satisfies the visual spatial and gives a 3D experience that when heard can also be felt, depending where the audio was recorded. Resnet in simple terms is a residual neural network which is an artificial neural network of any kind that would build on constructs known from pyramidal cells into the cerebral cortex. Sounds cool right? with Resnet this would be done by utilizing skip connections and short cut to jump over layers. In more complex terms along with resnet there is VGG-19 and feedforward neural network. For example, resnet 152 will have 152 layers of convolution where there is a subtraction feature learned from input of the layer by using shortcut connections directly connecting input of the nth layer to some n+xth layer. Being proven that residual learning can improve performance of model training, especially when the model has deep network similar as this experiment having more than 20 layers. With 12 total outputs from ResNet-152 and VGG-19 were used as input in a two hidden layered feedforward neural network. With the final outputs computed through the hidden layers. With the numbers at each layer representing the number of nodes on each layer. As expounded on resnet 18 block is two layers deep, with a total of 20 layers.

In figure 10 when passing the mixed mono files, they lead to STFT. Envelope Distance is the direct comparison of raw waveforms that may not capture perceptual similarity well. In this experiment they take the envelope of the signals and measure the Euclidean distance between the envelopes and ground truth left and right channels with predicted signals. The 1x1 convolution as done in the diagram reshapes the visual feature tile which is an ImageNet or pretrained resnet 18 network to extract visual features. With the audio files U-NET is used to extract audio features into joint audio-visual analysis to get the correct output. The U-NET encoder decoder network adopted is ideal for the dense prediction task where the input and output would have same dimensions. The difference of two channel can be predicted by taking ISTFT equaling to left and right monaural audio recorded. The mixed audio of the left and right recorded audio. Where we use this to try to predict the complex valued spectrogram for ISTFT.

Predicting this information into a complex mask which in turn will take the previous STFT and resnet-18 and develop it into ISTFT. Giving the predicted two channel binaural audio output satisfying the visual spatial configurations. U-NET as mentioned above is convolutional network architecture for fast and precise segmentation of images.

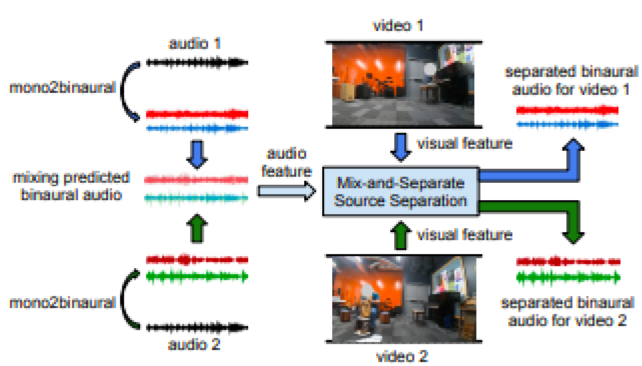


Figure 12: Audio to Visual

The figure above illustrates the separation framework. We mix the sounds of the predicted binaural audio for the two videos to generate a complex audio input signal. With the objective being to separate the binaural audio for each video for the specific visual frame. This is to obtain large gains by inferring binaural audio. The inferred binaural audio will give more informative about the audio compared to the original mono audio recorded, making for a cleaner separation. To most to be able to generate a sound from a silent video is absurd but due to technology, from material properties revealed by the sound's objects make when hit with a drumstick can be used to analyze new sounds from mute videos. For recurrent networks and conditional generative adversarial networks can at times generate audio for input video frames. Powerful simulators can also sync audio and visual data for 3D images. Rather than generating audio from nothing, converting an input of one channel audio to two channel binaural audio guided by visual frames. Making the system synthesize sound from a speaker in a room as a function of viewing angle and also assuming access to acoustic impulse recorded in any location.

Convolution was heavily used in this approach, specifically the 1 x 1 convolution. Which maps pixel input with all channels to output. This reduces the number of channels but is very slow due to the number of large depths. Analyzing visual imagery is done by using deep convolutional neural. These networks have applications in image and video recognition, and processing. Done in the figure above source separation audio takes cases where only single channel are available. Due to separation being easier when multiple channels are observed using multiple microphones or binaural audio as done here. From here transferring mono to binaural when observing video and leveraging the resulting representation to improve audio visual separation.

	FAIR-Play		REC-STREET		YT-CLEAN		YT-MUSIC	
	STFT	ENV	STFT	ENV	STFT	ENV	STFT	ENV
Ambisonics [28]	-	-	0.744	0.126	1.435	0.155	1.885	0.183
Audio-Only	0.966	0.141	0.590	0.114	1.065	0.131	1.553	0.167
Flipped-Visual	1.145	0.149	0.658	0.123	1.095	0.132	1.590	0.165
Mono-Mono	1.155	0.153	0.774	0.136	1.369	0.153	1.853	0.184
MONO2BINAURAL (Ours)	**0.836**	**0.132**	**0.565**	**0.109**	**1.027**	**0.130**	**1.451**	**0.156**

Figure 13: Statistics

Displayed above showcases the differences between where the audio recording was taken. With the lower the number being the better or more clear designation. For the experiment to test how the environment felt they had 18 participants with normal hearing. Ambisonics is where being a more general audio presentation is ideal for video, with predicting first and then decoding binaural audio to introduce artifacts making the binaural audio even less realistic. For e FAIR-Play dataset. The dataset contained level sounds of diverse sound making objects being good for mixing and separating the audio and visual. Monaural audio has very limited dimension. Due to collapsing all independent audio streams to the same spatial locations of sources in sound. MONO2BINUARAL deep convolutional neural network converts mono audio to binaural audio by inputting spatial facts in visual frames. The conversion process helps audio visual separation. Which separates mixed mono audio into sound sources relying on spatial facts in visual frame.

Converting monaural audio to binaural audio and leveraging video frames creating MONO2BINAURAL deep network to achieve this output. Then collecting enough fair play is evident to set with the audio gained for research of the audio and video dataset. Performing source separation and confirming the sound source.

With Mono2Binaural Network taking the mono audio and frames as input and predicting the output. This extracts frames from the center using resnet. So, to pass these visual features through 1×1 convolution layer and reduce the channel dimension it needs to be flattened and into a single visual feature vector. Spectrogram masks is proven better than alternative due to the direct prediction of spectrograms and raw waveform. In all this had many variables similar to ours, which resembled many of the ideas we had when brainstorming our design. The technology aspects as well as the engineering behind it played a major role in why we saw this experiment like ours.

4. Related Standards and Design Constraints:

The following section will outline many of the design standards and related constraints. Design standards are documents/guidelines that guide the designer in procedures, specifications, and health considerations (if any). The reason why these standards exist is so that companies in the industry can all follow the same set of rules and guidelines in product design and to make sure all products are designed in a safe manner. For our design, we have many standards that we are going to strictly adhere to. The standards such as LED Safety, C++ and ultrasonic testing will lay a critical foundation to our design.

4.1 Related Standards:

Standards are fundamental for any design. Standards help lay the foundation of projects and allow projects to be compatible with many devices. An example of a popular standard is the USB or WiFi. Both technologies are interchangeable with many devices. The standards that our project will be gathering from will come from IEEE Standards Association which help lay the foundation for many of the standards we see today. IEE, or the Institute of Electrical and Electronics Engineers, is a technical engineering organization that keeps up to date with emerging technologies in the electronics industry.

Furthermore, we will take into consideration ISO standards. ISO standards are some of the most known standards in the industry and are used all over the world.

With a combination of the plethora of standards from many reputable organizations around the world, it will be ensured that hazards are prevented (or at best kept to a minimum) and many of the regulations involving our project will be enforced.

4.2 Soldering:

ISO 12224: Describes the standard when it comes to soldering. Since we will be soldering, the type of flux to use, and how to properly apply the flux. Furthermore, soldering is a dangerous procedure, many things can go wrong and you can hurt yourself or others around you. Since we are soldering thousands of LEDs together, this standard is important so that we will ensure that we are applying the flux properly and soldering safely and efficiently. We want to ensure that we are soldering in the safest environment possible. We also want to ensure that any safety precautions we can take to minimize danger, we take. By following this standard our LED Cube will be soldered efficiently and take less time than if we weren't following this standard.

4.3 Programming Languages – C++:

ISO/IEC 14882 is the standard that deals with the programming language C++, it's syntax, implementation, and form. C++ is a programming language based on C and expands on many of the things found in C. Things such as data types, classes, templates, and more that were found in C are expanded upon in C++.

Regarding our project, this standard is important because we will be using C++ to integrate the software with our hardware. Furthermore, it also defines proper I/O, and data processing which we will heavily use in our project since we need to process the information received from the ultrasonic sensors and output a 3D Image.

4.4 Design Impact of Using C++:

In order to comply with the standard, we will take every step to follow the guidelines listed to ensure that we meet the standard. Furthermore, doing so will allow us to be able to easily show our code to other engineers and because it complies with those standards, they will be able to discern it more easily than if we didn't comply with the standard.

The reason why we chose to use C++ vs any other programming language is because of C++ 's portability, mixed in with how the majority of Bela's functionality is designed in C++. Because of this, all aspects of the software design will be implemented in C++ and will therefore use this standard.

4.5 Ultrasonic Testing:

ISO 5577:2017 is the standard that deals with ultrasonic use/testing. This standard particularly deals with non-destructive and general use. It is important that we implement this standard because the bulk of our project relates to the use of ultrasonic sensors. It describes the basics of how ultrasonic sensors work, and furthermore how to implement ultrasonic sensors for general use. By implementing this standard, we make our ultrasonic sensors safer, more efficient, and more standardized.

4.5.1. Design Impact of Ultrasonic Testing:

This standard is important to our project because along with helping us formally implement a standardized technique in building our array of ultrasonic sensors, it gives us a baseline on how to implement them, the main thing this document does is to inform the reader on all the techniques available in using ultrasonic sensors. Techniques such as pulse-echo technique (where an echo is sent out and then received before the next echo is sent out) and the tandem technique (where two probes are angled in such a way that one can send out the echo and the other can receive it) will be helpful techniques in deciding how to formally set up the array of sensors.

4.6 LED Safety Standard:

IEC/EN 62471 is the standard regarding LED safety. Specifically exposure limits and measurement techniques. Furthermore it goes on to mention the various biological hazards that go along with certain wavelengths. For example, for a wavelength of 315-400nm there is UVA eye hazards associated with that wavelength. Along with that, there are classifications for the hazards. There are 3 groups, low-risk (group 1), moderate-risk (group 2) and high risk (group 3).

4.6.1 Design Impact of LED Safety:

IEC/EN 62471 is potentially one of the most important standards we will implement in our project. This is since safety must be a top priority when doing any project and more important than a project working is if it's safe. Therefore, we will strictly follow these guidelines when building our LED Display. We have to make sure that our project is as safe as possible, such that when we display our project in the showcase it will not harm anyone.

4.7 Realistic Design Constraints:

In every system, design constraints are formed. This is due to the project being limited by political/economical/time conditions. Constraints are restrictions in the project that limit the number of things you can do. In our project, we were mainly limited by economical and time conditions, however there are many other constraints that should be analyzed as well. Furthermore, analyzing the constraints can give us a good idea on where we can direct our project and if possible We begin to look at these constraints individually, and then we will look at how each constraint is connected to each other and to the overall design of our project.

4.7.1 Economical and Time Constraints:

Due to economic constraints on our project, our part selection will have to be limited. Certain features may have to be removed or limited and certain desired parts may not be able to be included. Currently, our budget which isn't sponsored by anyone is $1,000. We should be able to reach below this. The costs of the Bela Board, the sensors, the mounting mechanism, etc. easily fits below this. However, we may have to choose cheaper materials for certain things in order to save money.

Time restrictions should also be considered. Carefully planning who does what and what time frame which things need to get done in what order is essential to the project being done in a timely manner. Currently, we have the research/planning/testing phase of the design set to be completed by the end of Fall 2019. Afterwards, we plan on beginning the building process for the design. This building process will be completed by the end of Spring 2020 to participate in the Senior Design showcase. During this phase, we will test rigorously to rule out any errors with our design process. Time constraints are important because compromises might have to be made, as an example, if a feature to implement takes too much time to implement, we might have to remove it in order to focus on everything else.

4.7.2 Manufacturability and Sustainability Constraints:

Constraints regarding manufacturability involve constraints that limit how our design can be manufactured. We are constrained due to the fact we only have the tools and materials available to us and this goes hand in hand with the economic constraints. Furthermore, sustainability constraints involve constraints that impact how long our design lasts. To this end, choosing materials and building methods that are durable and made to withstand many environments/conditions is optimal. This would also include what mounting mechanism we choose to implement for the sensors, the LEDs and what materials we choose to connect everything together. Furthermore, there is usually a negative correlation between manufacturability and sustainability. Therefore, choosing a "sweet spot" where we can achieve the maximum of both is of the utmost importance.

4.7.3 Environmental and Safety Constraints:

Constraints regarding safety include making sure that all of the components are compatible and will not be dangerous to move/operate. There are many instances of projects catching fire, exploding, or injuring the creator and user. Luckily, our project is relatively safe, and we do not predict anything going wrong, however we will take every precaution necessary. Furthermore since our project is being shown on the UCF Campus, every step to ensure that it is a safe project is being taken by UCF.

Another potential area of concern would be how the LEDs affects your body. Since we plan on using a lot of LED's we must ensure that prolonged exposure to LED's will not cause any negative side effects to your health. Much research has been done regarding that and thankfully, it has been shown that LED's are not harmful to your body. Not only that, but we must ensure that the electrical components we choose are able to be recycled and are environmentally friendly.

Safety is a constraint that absolutely cannot be ignored. Therefore, it is imperative that we take the necessary precautions in order to ensure the best user experience, with the best functionality, while also maintaining a high degree of safety.

For example, "Safety constraints typically include things like requiring interlocks and physical barriers around moving parts, safeguards concerning electricity and the handling of toxic chemicals, and the mandatory placement of warning signs. A potential danger in the mandating of specific safeguards is that it may well be possible to architect a better system in which the associated hazards cannot occur and thus the mandated safeguards become unnecessary or inappropriate. In fact, the new system without the safeguards may be both cheaper and safer. For example, using magnets to keep refrigerator doors closed eliminated the need for installing safeguards to allow trapped children to open the previous locks from the inside." (The Journal of Object Technology)

In line with ethical and safety constraints are health. Health constraints focus on the health of the consumer, designer, and ecosystem affected by a specific device. If a device is causing harm to the environment, causes a sickness, and or kills a living organism is it a health concern. Also being anything, which impedes the ability to provide and or deliver care due to interference.

According to "Energy consumption has moved to center stage of disquiet about the global environment and economy because of concerns about the effect of current systems of energy production, the galloping growth of energy consumption (IPCC 2007), and the growing global competitive demand for energy resources. Macro policies, such as the Kyoto Proto- col; regional policies, such as the European Union cap-and- trade carbon trading system; national and regional policies, such as state-level renewable energy portfolio standards (U.S. Department of Energy [U.S. DOE] 2008); and the introduction of various energy use and management options targeted at end customers by power companies, are indicative of the market system–wide recognition of the twinned global concerns of environment and energy." (Nadir Yurtoğlu - History Studies International Journal Of History - 2018)

4.7.4 Ethical, Social, and Political Constraints:

When it comes to large scale design projects, ethical, political and social constraints must be looked at. A common ethical issue when it comes to technology that does 3D imaging is privacy and how these technologies can potentially invade privacy. This issue is also a privacy and social issue since recently many companies have been under criticism for selling user information to advertising companies. That's why it is imperative to let the user know what information is being stored and what information isn't. This will allow the user to consent to give up privacy/know what information of theirs is being used and for what purpose. Further ethical constraints involve issues such as can our project be used to harm others. Can it be used to do morally unjust things? The answer to these things (as far as we know) is no, however we should be mindful of them.

Another constraint that might seem odd, are pricing constraints. These constraints fall in line with social constraints. Pricing constraints usually occur when the product is way too expensive. For Example, pricing constraints usually are the things that keep people from having flexibility in the pricing decisions made when marketing a design. So ceiling prices come in to play set by customers and even companies depending on budget. Where companies take into effect a break-even point. This has a major effect on what customers are willing to pay and the floor which is where a company breaks even with how much money and time they put into designing an object.

Political Constraints usually correspond to taxes and subsidies. Including different policies that may be in effect.

For example, “Regulatory constraints typically take the form of minimum or maximum quotas on the process or any of its exchanges. Wild-catch fishery is an example of an activity constrained by quotas world-wide, reflecting an underlying resource scarcity. Product quotas are also common within the EU agricultural policy, where e.g. such products as milk and sugar are regulated by quota systems, as an indirect income subsidy for the farmers. Taxes and subsidies may also constitute constraints on specific producers. An example is the negligible import of cereal grains to the EU, because of a very high import tax. Similarly, the farmers’ choice of crops is strongly dependent upon the level of subsidies given for different crops, virtually imposing a constraint on crops less subsidized. Quotas, taxes or subsidies that are exclusively politically motivated, and not reflecting an underlying resource constraint, are often temporary and may in some situations be influenced by changes in demand. It is therefore recommended to apply a sensitivity analysis to constraints based on such political regulation.” (Irimia R, Gottschling M (2016) Taxonomic revision of Rochefortia Sw. (Ehretiaceae, Boraginales). Biodiversity Data Journal)

5. Relevant Technologies

Like Max Cooper's installation, we could've opted to use reflective beads and projectors to simulate 3D images. This would've been harder to do on a small scale but potentially more adaptable to use on a bigger scale. We could've also opted to use a laser system as a scanner instead of ultrasonic sensors. This system would've been much easier however it would've defeated the spirit of the project. We also could've built a 3D representation using a game engine such as unity in a Virtual Reality and react with objects in the real world but in Virtual Reality. Furthermore, the underlying adaptation of the Frame for the ultrasonic sensor array would make a good basis for a night vision system.

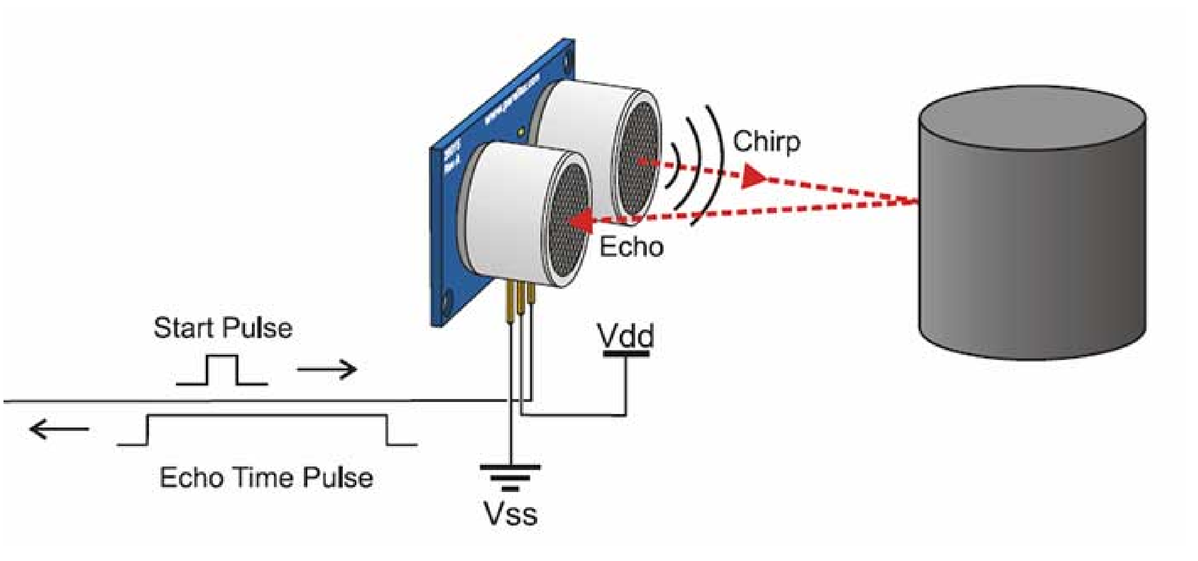
Relevant technologies needed for us to invent this design range from physical to nonphysical aspects. The first would be programming or specifically C programming. C programming created in 1972 by Dennis Ritchie at bell laboratories at AT&T. Is allowing us to create a code so that we can have the LED do as instructed from our program. If C programming, or programming in general was not created there would be no way for us to tell our hardware what to do and how we want it done. Since we can talk to the hardware, we are able to instruct it to do certain commands by simply creating a code to translate what we want done in machine language to speak to our hardware. For us to comply with the standard, we are taking every step to follow the guidelines listed to ensure that we meet the standard. Furthermore, doing so will also allow us to be able to easily show our code to other engineers and because it complies with these standards, the other engineers will be able to discern it more easily than if we did not comply with these standards. Our reasoning for choosing C++ instead of the other programs is because with C++ portability, mixed with how Bela's functionality is designed in C++. This always aspects of the software design to be implemented in C++ and will therefore use this standard and a smooth transition.

We also chose to use C++ because C++ is extremely portable and a standard when it comes to programming. As a result, it has a lot of support from the community and is easily able to be transferred to different systems without any worry. This will become extremely useful if anyone else were to ever follow this paper and recreate the project.

Ultrasonic are also a very big part of our design which was already created. Ultrasonic is relevant due to the fact ultrasonic always for our arrays to create the environment needs for our design. With ultrasonic we would have to find a different way to detect and display our design. helps us formally implement a standardized technique in building our array of ultrasonic sensors, by giving us a baseline on how to implement them. The main thing this does is to inform the reader on all the techniques available in using ultrasonic sensors. Techniques such as pulse-echo technique (where an echo is sent out and then received before the next echo is sent out) and the tandem technique (where two probes are angled in such a way that one can send out the echo and the other can receive it) will be helpful techniques in deciding how to formally set up the array of sensors.

Once decided, we will build the frame in order to securely mount the sensor arrays. This will allow the sensors to sit still in order to properly read input/output.

With *ISO 5577:2017* being the standard that deals with ultrasonic use/testing. This standard particularly deals with non-destructive and general use. It is important that we implement this standard because the bulk of our project relates to the use of ultrasonic sensors. It describes the basics of how ultrasonic sensors work, and furthermore how to implement ultrasonic sensors for general use. By implementing this standard, we make our ultrasonic sensors safer, more efficient, and more standardized.



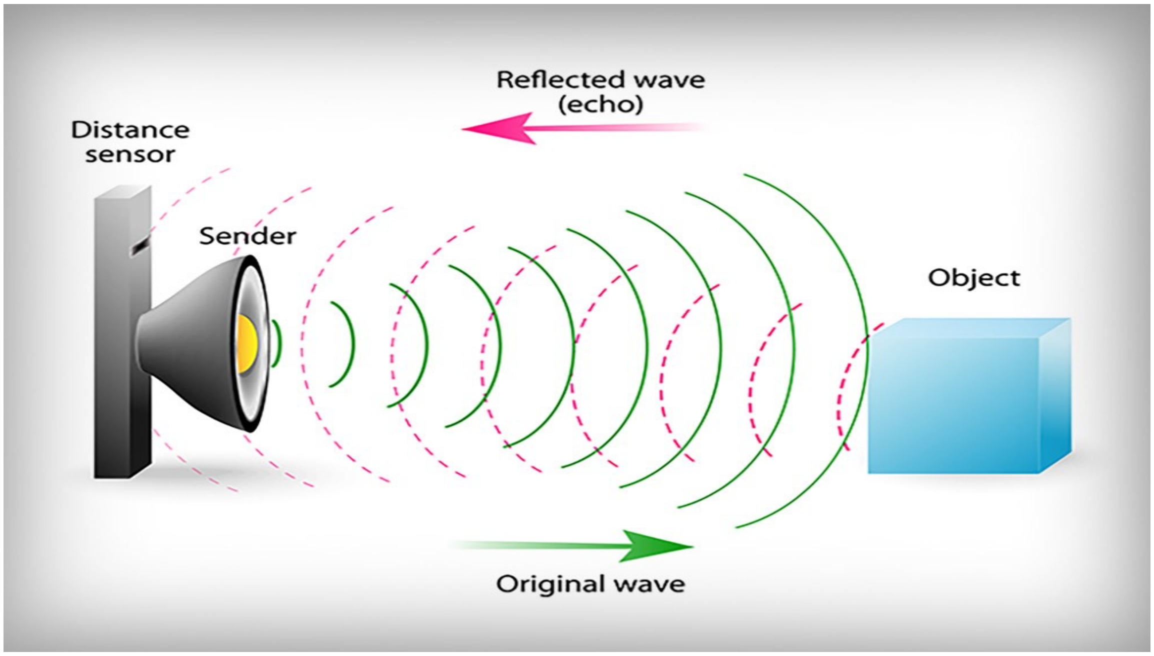


Figure 14: How Ultrasonic Sensors Work

Another relevant technology is the LED. I covered LED slightly before which stands for Light-emitting diode. With this solid-state lighting which uses semiconductor to convert electricity into light. The small design is idea for our design not to mention how small and efficient it is at producing light in one direction. Since we got our inspiration from watching artist perform and all of the vast and colorful lights displayed in their performances. LED became very popular being used in Television, Entertainment clubs, and many other appliances. Being the most efficient lights on the market these lights were a go to for us not to mention all the color variety. For our LEDs, we want LEDs that can be equally visible from each side. In this regard, we will use diffused LEDs since they are equally bright from all sides. Furthermore, we need LEDs that can be easily mounted as well so that when displaying our design, the LED won't fall due to the vast amount we are using. So, we will use circular LEDs since they are easier to mount versus square LED. Since we are using in LED a large quantity, an optical illusion to create the appearance of a 3D image, by flashing each layer on and off quickly we will need LEDs that are bright enough to produce a good amount of light in a short time period that the LED will actually be on. Otherwise, the image will not look right. Lastly, the legs of the LEDs must be all the same length since we are going to use the legs to help mount the LEDs. We will also test every LED to ensure that they function properly and are not damaged or not working properly. These being all the relevant technology used in our design for our visual display system.

6. Hardware Selection and Analysis:

In this section we will thoroughly go over the hardware components we selected, including pros/cons of all our choices as well as describe a lot of the design decisions used to make these choices. Furthermore, we will compare how these technologies work, and which will fit best for the purpose of our project.

6.1. HC-SR04:

In terms of Ultrasonic range finder devices, the HC-SR04 is probably the most ubiquitous design available. Two transducers connected to an 8-bit microcontroller allow for an easy and cheap method of detecting range. In lay terms the module sends out a high frequency "chirp" through one transducer and receives the echo generated by said chirp at the second transducer. When connected to a microcontroller/microcomputer one is able to write a small amount of code that takes the time between pulses sent from the output transducer (Trig) and those received at the input transducer (Echo). Using this time and multiplying by the speed of sound in air we are able to get the round-trip distance of the chirp (and thus the distance of the object which reflected the initial chirp).

The HC-SR04 has been chosen for its well documented characteristics (operating ranges, cone of effect, ETC), wide availability, and low cost. With a +/- 3mm tolerance the device should be well suited for imaging of larger figures. Additionally, the package itself comes with four mounting holes for screws that allow them to easily be installed into the grid required by the Frames input system. These modules can be found quite cheaply from international suppliers and can also be made readily available domestically, albeit at a cost that may be prohibitive to install at a large scale.

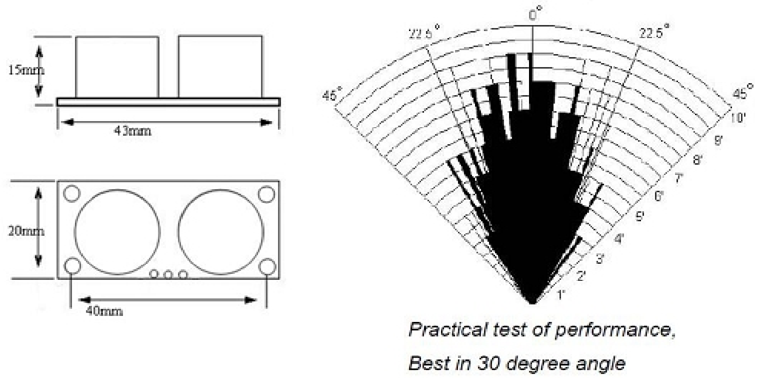


Figure 15: Range of Ultrasonic Sensors

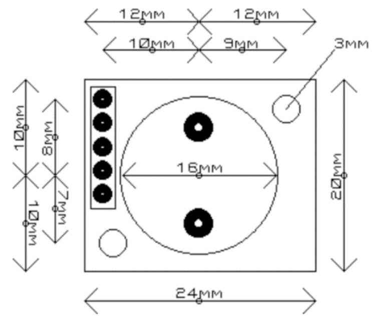
While the HC-SR04 excels in form factor, price, and reliability (in both operation and sourcing), it does have some potential issues when being installed in the Frames unit that must be considered. As the module uses sounds to determine distance, cone propagation begins to affect the reading. A cone is projected in front of the Trig transducer and anything within that cone is subject sending an echo back to the Echo transducer. Fortunately, there is correction system for this potential data fuzzing, the HC-SR04 only records the data of the nearest (shortest time between pulses) object. While this works quite well for a single sensor set up at virtually shrinking the cone of "vision" of the unit, the effects of sound propagation begin to become a legitimate problem when one considers stacking upwards of 256 units together. As the units are placed side by side an issue quickly becomes apparent that the Echo Transducer of one module is extremely close (and potentially within the cone of effect of the transducer itself). As such it becomes important to think of a way of limiting crosstalk.

One possible method of "shrinking" the cone of effect is to erect dividers around each Trig transducer. This will effectively clip the propagation and allow more space for the echo to receive its intended chirp as opposed to that of a neighbouring module. Each module has a cone of vision of about +/- 30 degrees from the center. And as such it may be possible to restrict that cone by simply erecting walls on either side of the module. These walls should be made in a way that they will not cause extraneous reflections from hitting the Echo transducer or else all of the work put into limiting crosstalk may be wasted as the dividers may generate false readings if they are more reflective than need be. Potentially placing dividers around only the transducer may remedy this as the transmission range will be shrunk and the Echo transducer will not be subject to an overage of excessive reflections, thus avoiding extraneous readings from being generated.

6.2 SRF02:

Another worthy contender for our sensor of choice is the SRF02, this is a single transducer range finder in a package less than half the size of the HC-SR04. The SRF02 boasts a slightly tighter "beam" shaped dispersion pattern than its dual transducer cousin as well. This allows the SRF02 to be more easily installed in our system as its square shape will allow for a more accurate and denser image capture.

Additionally, the SRF-02 has fantastic I2C support and there are libraries developed for running an array of them, aiding our software development process immensely. These reasons coupled with the sensors significantly lower current draw (4mA vs the HC-SR04's 15mA) make the SRF-02 a potent choice in sensor for our project.



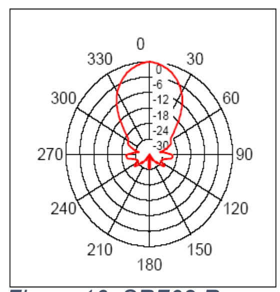


Figure 16: SRF02 Range

One of the biggest advantages the SRF02 holds over the HC-SR04 is its shape and size. The HC-SR04 is a rather bulky device, its overall rectangular shape does not lend itself well to emulating the fine voxels of the LED display that will be outputting its information. The SRF02, on the other hand, is nearly a square, and can be tightly packed together to better increase sensor density.

As image clarity is largely a function of pixel density (ie cameras sporting resolutions of several Megapixels per unit area and screen displays packing more and more pixels into a given space) sample clarity also increases with a higher density of sensors. In short, the same rules that apply to image reconstruction also apply to image capture. As the goal of our system is to directly map the image of what is in front of the Frame to the LED Cube, being able to increase overall density of the sensor would better increase the image reconstruction.

This benefit in shape is especially apparent when one considers the decreased minimum operational range. The major limitation in operation range if we opt for the HC-SR04 comes from its shape. As the HC-SR04 is rectangular in form, its center point (and therefore its optimal imaging point) cannot be as close to other modules center points as that of the SRF02. This causes the "mesh" of vision to be rather spread out when relatively close to the input array. By increasing the overall density (only obtainable through opting for a package smaller than the HC-SR04, for example the SRF02) would tighten said mesh and thus produce a more lifelike image when reproduced. In theory the gaps made by the sensor spacing could be filled/interpolated by software-based edge detection programs, it would be an overall simpler and more efficient task to increase sensor density if we are financially able to.

While the beam pattern seems to be tighter, and the smaller more usable size and shape are an advantage over the larger HC-SR04, the SRF02 has a handful of caveats that prevent it from becoming an obvious choice. The smallest of these caveats is its minimum operational distance. As it is a single transducer rangefinder, the SRF02 does not have a dedicated return input for echo, instead after each pulse is sent it listens for the echo using the same transducer. This means that there is a minimum wait time imposed on the unit between Send and Receive cycles (once a pulse is sent said pulses echo must be received before sending a new one). This will take a small hit to sample rate and also cause the SRF02 to have a much larger minimum operating distance than the HC-SR04. Specifically, 15cm vs the dual transducers 2cm. While the Frames system inherently requires some space between the target and the input array to properly reconstruct images, this increase in minimum operational distance must be taken into account.

The most daunting characteristic of the SRF02 is its cost. HC-SR04's are available domestically at around $1 to $5 a piece and internationally sourced in bulk for as few as $.50 apiece. The SRF02 on the other hand is available domestically for an exorbitant $13 apiece and $4 a piece internationally. If the goal of the Frames system was to make a small-scale texture mapping system this would be fine, but as we are developing a system utilizing a 16x16 input array even a few cents, let alone several dollars increases our build cost exponentially. As it stands a 16x16 array of HC-SR04's would cost $128 while the same sized array built using SRF02's would cost $1024. This would make the project prohibitively expensive in its current funding state.

In short, the SRF02 is a formidable choice, and potentially the best choice, in widely available ultrasonic rangefinder modules for our image capture system. Unfortunately, this quality of product comes with a steep premium that may be too high of a cost to properly justify installation in the current iteration of the Frames system.

6.3. HC-SR04 vs. SRF02 Break Down and Conclusion:

The table below shows a comparison between the HC-SR04 and the SRF02. It helps demonstrate why we chose the HC-SR04 vs the SRF02. This will allow the reader to clearly see our design decision and why we made the choice we made.

	HC-SR04	SRF02
Current Draw	15 mA	4mA
Size	20mm x 40mm	24mm x 20mm
General Shape	Rectangular	Square
Operating Voltage (VCC)	5v	5v
Pins	4	4
Best Operational Range	3cm - 4m	15cm-6m
Angle of Operational	≤ 30deg	≤ 30deg
Resolution	1cm	3-4cm
Min. Per Unit Cost Est.	$.57	$4.00
Per Frame Cost	$145.92	$1024

Figure 17: HC-SR04 vs SRF02

While the SRF02 may be better suited in terms of size, shape, current draw, and maximum operating distance (while also providing a higher image resolution due to the potentially increased sensor density), its steep cost prevents us from being able to go forward with utilizing it. The HC-SR04 is cheap enough to make up for its relative short range of sight and potential crosstalk issues, both of which can be designed to accommodate. Additionally, the HC-SR04's 1cm resolution will at a minimum generate a higher clarity of depth perception in our 2D representation of the data stream. In theory, given a sufficient power supply, this project can be constructed using either the HC-SR04 or the SRF02 as they more or less the same operational patterns and the same pin set up. If down the road a financier were to support the project, swapping to the SRF02's would be a simple task of restructuring frames to accommodate the new shape and smaller overall footprint.

6.3.1 BeagleBone Black:

As the Frames system is attempting to process 256 continuous data streams, convert them for graphical representation, and output them directly to a 3D display, it stands to reason that it requires a large amount of computing power to guarantee all these computations are handled in real time. As the goal is to have fluid motion and fluid recording round trip latency (frame to cube delay) is chief among our concerns. If a fluid, near real time image cannot be generated much of the systems effects will be lost and some forms will be indistinguishable. To remedy this we have opted to use the BeagleBone Black platform. This microcomputer is well suited to our needs, sporting a 1Ghz ARM Cortex-A8, ample gpio pins, 512MB of RAM, and 4GB of flash memory. With this much headroom running into issues regarding resource shortages should be mitigated, thus guaranteeing (from a hardware perspective) enough computational resources to facilitate real time 3D imaging. Additionally, the BeagleBone Black's open source nature has lead to a large community of developers and manufacturers creating modifications and capes for just about any foreseeable purpose.

Additionally, the Beaglebone Black features a wifi chip that would allow for wireless interfacing with the data stream, and interconnect ability between microcomputers. While not directly applicable to our current system, if we are to increase the Frame's system beyond the scale of our current project, interconnectivity would become of paramount importance. With wireless communication between microcomputers, we would have the Frames input unit any distance away from the 3-Dimensional LED Cube Display. This would enable the system to be installed in almost any space that it would require, freeing it from tethers of wired communications.

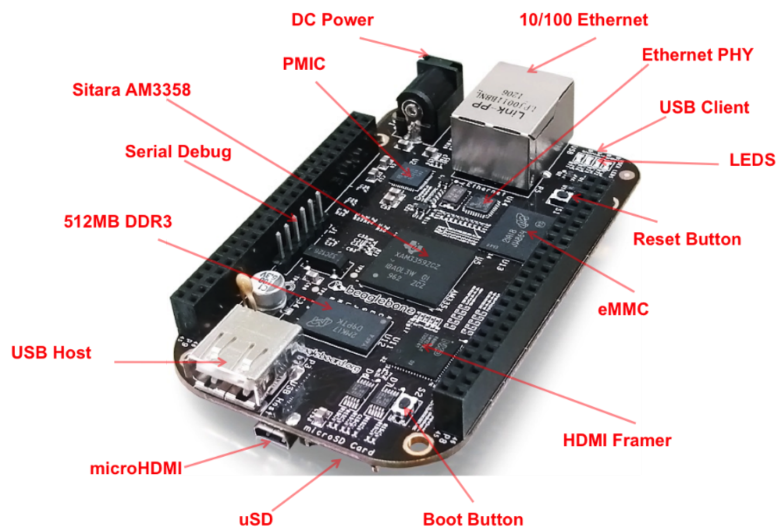


Figure 18: BeagleBone Black Hardware

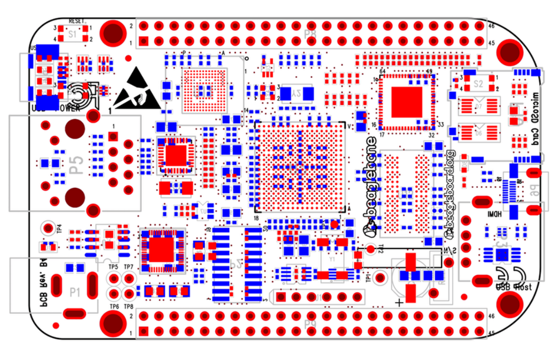


Figure 19: Beaglebone Black Schematic

6.3.2 Bela Cape:

One of the capes available for the BBB is the Bela. Initially made for the development of audio processing tools and general experimental instrument design, the Bela works as a fantastic modification to the BeagleBone Black especially when real time I/O is of the highest priority. One of the main advantages of using this cape in particular is its Linux shell that puts the highest priority on maintaining I/O speed and fidelity. Reportedly the cape can perform round trip I/O with times as small as 1ms, a speed ample enough to provide a responsive feel to the system. With the custom OS on board comes a portable IDE that allows for rapid prototyping and sensor data monitoring that would otherwise be cumbersome to develop on its own. The IDE has custom support for numerous languages, including C++, Super Collider, and Java. Below is an image of the Bela board as well as the Bela schematic.

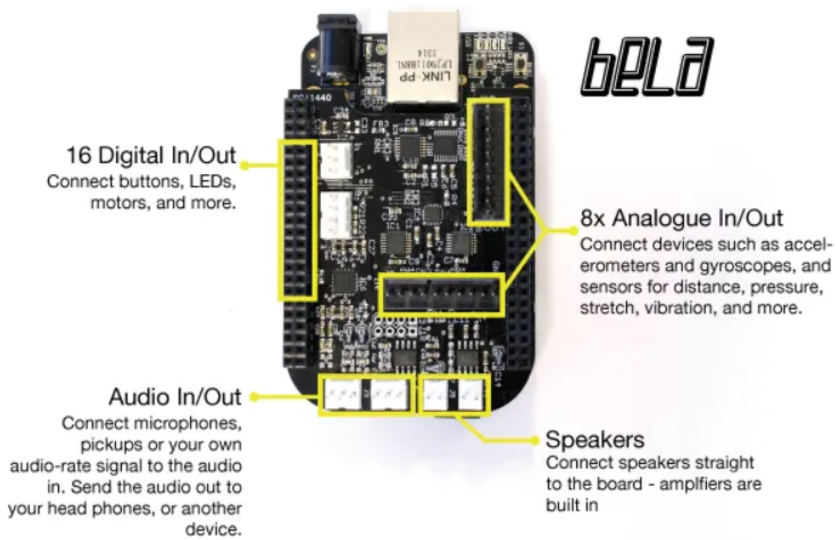


Figure 20: Bela Hardware

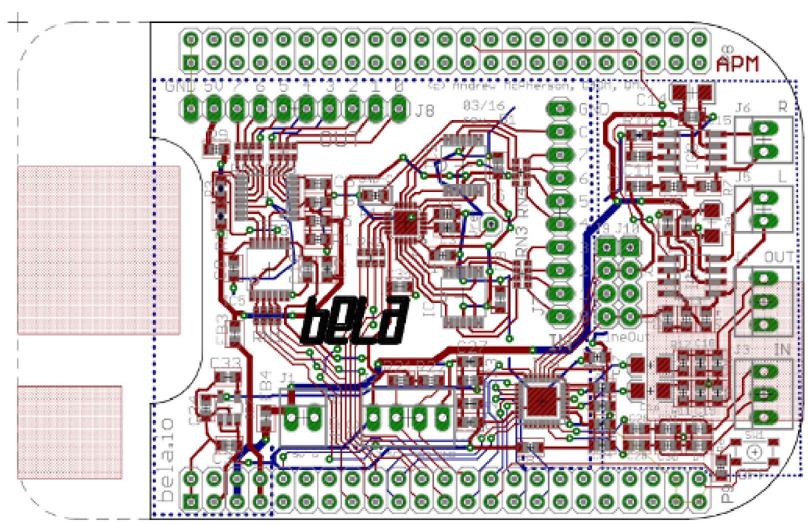


Figure 21: Bela Schematic

Due to many of the Bela designs being open sourced, there's a plethora of wealth online about how the Bela is designed, open-sourced projects for the Bela, and much more.

As it was initially developed for music and audio based embedded systems the Bela comes equipped with several analog and digital pins whose processing fidelity was designed to ensure the highest quality audio rate sampling. As we only need to sample and output at a rate of 24 frames per second (using film as a baseline for fluid motion) the Bela is more than well equipped to handle the sampling requirements of our current system.

While adept in the resource and sampling department to be able to actually process the incoming data streams, there is not enough raw I/O ports to be able to intake every frame directly at once. The Bela comes equipped with 8 analog inputs, 8 analog outputs and 16 Digital I/O pins. 8 analog I/O pins are not enough to directly handle the 256 sensors that are currently in place for the Frame unit (and conversely not nearly enough to handle the LED Cubes 3rd dimension). In order to solve this problem, we must make a decision between analog multiplexing, and a progressive scan-type system.

Analog Multiplexing would require a few multiplexing chips but would allow us to step down our 256 inputs into a serialization able to be fitted into the Belas inputs. The modulated data stream would have to be demodulated internally once in the microcomputer for it to be properly interpreted and processed for output to the LED Cube. While cleary implementable this solution requires more parts than potentially necessary and requires additional software to be written in order to deal with the modulated data stream.

Progressive scan, like that found in televisions is a method in which the incoming stream of data is serialy used from one end of the array (in televisions top left pixel downwards to the bottom right pixel) to another. In taking this concept and looking at the set up for the Frame unit we are able to draw similarities between each sensor taking in an input and each pixel of a tv displaying a certain value. Instead of simply taking in one sensor at a time at a high enough refresh rate to simulate real motion, we can take groups of sensors process their data while then cycling through the remaining sensors at a high enough rate to emulate fluid motion. As it stands this may be a simpler option as it would potentially minimize any analog multiplexing required as we could simply use sensor group sizes that would directly map to the Bela Outputs. Implementing this would require some form of selection process that would allow the Bela to send a pulse to the Frame in order to cycle through each group.

Incidentally, progressively scanning each sensor may be a feasible solution to the crosstalk issue mentioned in the section on the HC-SR04. By grouping sensors into sample groups that ensure that no sensor is sampled at the same time as a neighbouring sensors, we should be able to completely evade the issue of cross talk (SEE FIG 1). This would potentially remove the need for any dividers to be installed between sensors in an effort to reduce crosstalk. This benefit puts progressive scan over only multiplexing as it prevents error prone communication and solves the issue of having to few pins to work with.

As a bonus, Bela's IDE has fantastic support for developing GUIs. One of the features we want to include in the Frames system is a two dimensional representation of the incoming data stream to help demonstrate the process of taking the data stream from the real world to the 2 dimensional sensor array and back into a 3-D space when output to the LED Cube. Bela’s support and graphics power should make it a breeze to develop a 2d display system, additionally this display output is easily accessible over USB or Wi-Fi via a laptop and can then be projected at any scale to best demonstrate what's happening in the system in real time.

As is becoming apparent as the design process of the Frames system continues, having a multitude of accessible digital and analog pins is a must.

6.4 Other Hardware Considerations.

The main purpose for choosing these two components over other boards such as Arduino or Raspberry Pi is because these two boards have enough processing power to handle the 3D imaging. 3D imaging requires a lot of processing power; therefore, we have to choose a microcontroller that has enough power to handle the amount of inputs there are, and the amount of outputs there are going to be. First we will directly look at the Arduino and the Raspberry Pi then look at both of them in comparison to the BeagleBone Black.

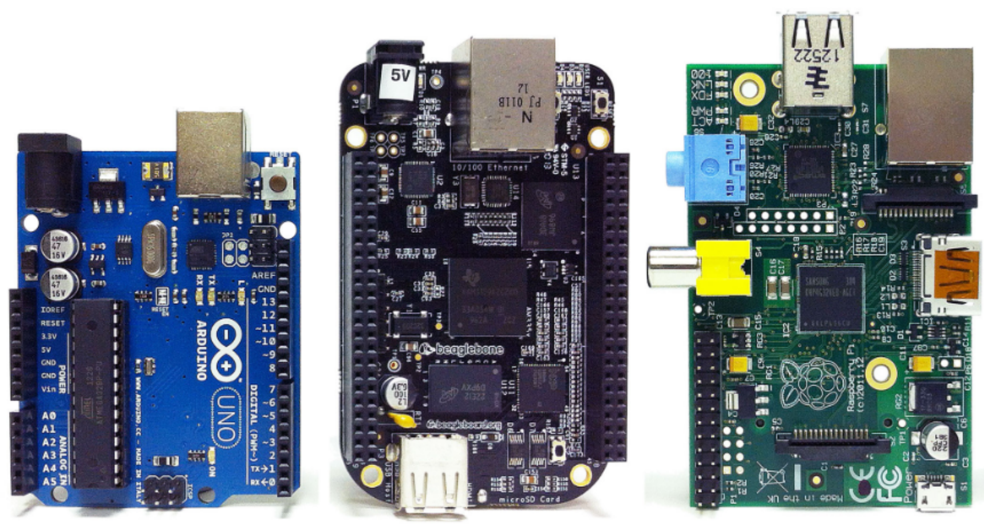


Figure 22: Arduino, Beaglebone Black, Raspberry Pi

6.4.1 Arduino

The Arduino is simple, and cost effective, however compared to the other two, it has about 40 times less the clock speed. In our application this would prove to be very detrimental. The Arduino also does not have a lot of computing power compared to the other two. It is also not suitable for complex projects due to the lack of video output and audio output.

6.4.2 Arduino

The raspberry pi is the middle ground. It has a good amount of computing power, with a cheap affordability and a good amount of memory, however it is not adaptable. Furthermore, connecting it to external sensors is difficult due to it only having 8 GPIOs. Lastly, compared to the other two, the setup is a little bit more tedious.

Due to these factors, it seemed as though the BeagleBone Black was our best bet to use. The computational power mixed in with the easy portability and the memory it had made it a powerful piece of equipment that would be able to serve us in our objective well.

Furthermore, when considering other forms of hardware to use, it was important to minimize the cost, in our case, the BeagleBone Black and the Bela Cape were both free since we had it on hand already. This saved us a lot of money and time because it allowed us to prototype immediately and not wait for anything.

One of major factors when it comes to real-time image rendering involves latency. To this extent, the Bela is miles above its' competitors. Along with that, the Bela can take analog inputs whereas every other microcontroller has digital inputs.

These are the main factors when choosing the Bela Cape and BeagleBone Black. Due to these factors we believe it is by far the best option to use to process analog inputs with low latency. This will allow the images to be processed smoothly. Listed below are the hardware specifications for the BeagleBone Black vs the Arduino vs the Raspberry Pi. This table allows the reader to quickly compare many of the different qualities of each microcontroller. The orange highlighted is the one we chose.

	BeagleBone Black	**Arduino**	**Raspberry Pi**
Price	$89	$35	$29.95
Size	3.4" x 2.1"	3.37" x 2.125"	2.95" x 2.10"
Processor	ARM-Cortex A8	ARM11	ATMega 328
Clock Speed	700 MHz	700 MHz	16 Mhz
RAM	256 Mb	256 Mb	2 Kb
Flash	4 Gb (Micro SD)	SD Card	32 Kb
EEPROM	-	-	1 Kb
Input Voltage	5v	5v	7-12 v
Min Power	170 mA (.85W)	700 mA (3.5W)	42 mA (.3W)
Digital GPIO	66	8	14
Analog Input	7-12 bit	-	6-10 bit
PWM	8	-	6
TWI/I2C	2	1	2
SPI	1	1	1
UART	5	1	1
Dev IDE	Python, Scratch, Squeak, Clout9/Linux	IDLE, Scratch, Squeak/Linux	Arduino Tool
Ethernet	10/100	10/100	-
USB Master	1 USB 2.0	2 USB 2.0	-
Video Out	-	HDMI, Composite	-
Audio Out	Analog	HDMI, Analog	-

Figure 23: Hardware Comparison

6.5 LIDAR vs Ultrasonic :

Lidar stands for Light Detection and Ranging is a method to sense remotely by using light in the form of a pulsed laser to measure ranges to earth. These light pulsed lasers give precise three-dimensional information about Earths shape and characteristics of the surface.

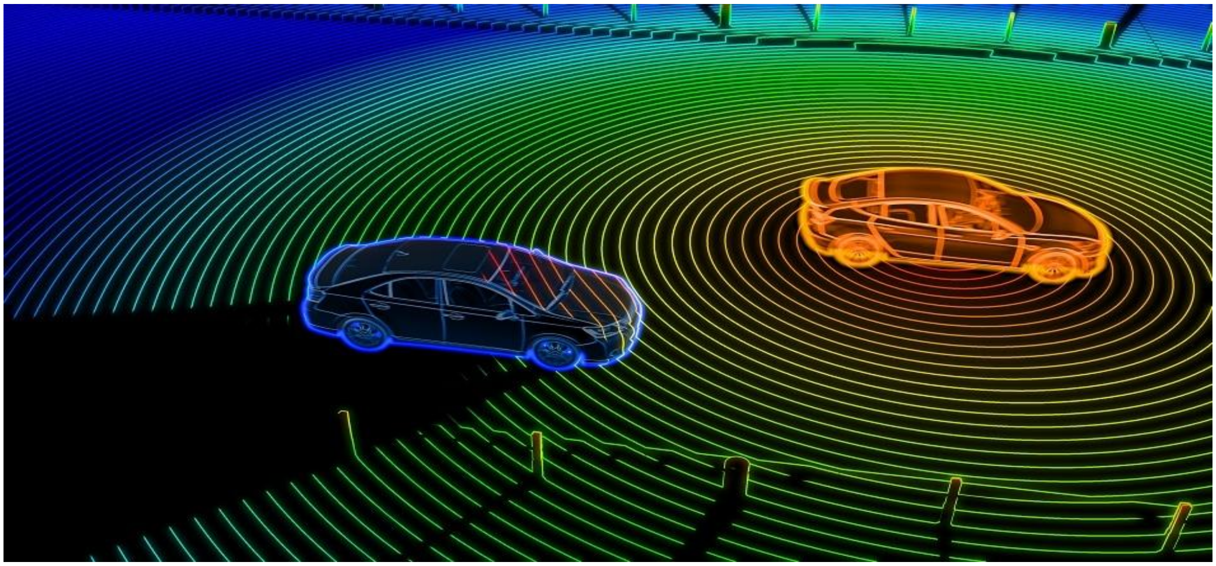


Figure 24: Lidar Light Detection Ranging

Lidar system usually consist of a laser, special GPS receiver, and a scanner. Objects that mainly utilize Lidar air mobiles such as helicopter and Airplane, which use Lidar to show information over broad areas. Lidar has two types topographic and bathymetric. The difference between the two is that topographic uses near infrared laser to map land while on the other hand bathymetric uses water penetrating green light to measure water depths such as seafloor and river elevation. These Lidar devices allow for scientist and professionals who use mapping to study manmade and natural landmarks and environments with precise accuracy. Many scientist use Lidar to produce accurate maps for shoreline and make digital elevation model to use in geographic representation. These maps assist in emergency response in the occurrence of environmental needs.

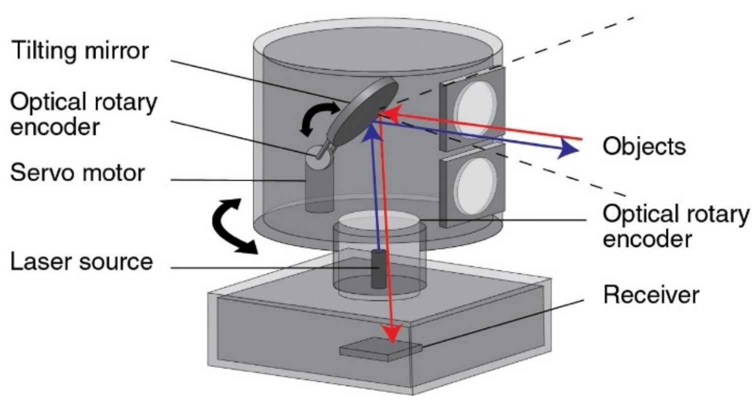


Figure 25: Motor for Lidar

Lidar differs from sensor array, radar, and camera in many ways. Camera works by using a lens which is a curved piece of glass or plastic to take beams of light bouncing off of an object and redirect it, so they come together to form an accurate image of what is present in front of the lens. This seems complicated but is done as light travels from one space to another changing speeds. Since light travel quicker through air than glass it slows the lens. Since light waves enter the lens at an angle it will leave one wave contacting the lens before another casing that wave to slow first. This bends the direction, so all rays converge to a single point. At that single point is when a real image will show. From this process a visual representation of what the camera is looking at should be displayed.

Radar stands for Radio Detecting and Ranging. Similar to Lidar which stands for Light Detecting and Ranging. Radar uses radio waves, by sending out electromagnetic waves as short pulses reflected by objects in the path back to the radar. Very similar to bats sonar or even an echo to give depth of the objects in the path. Radar consist of various instruments to complete its design. A transmitter to create energy pulse. A transmit and receive switch to communicate with the antenna. An antenna to send pulses into the environment and receive the pulse back. And lastly a receiver to detect and transform received signal into video.

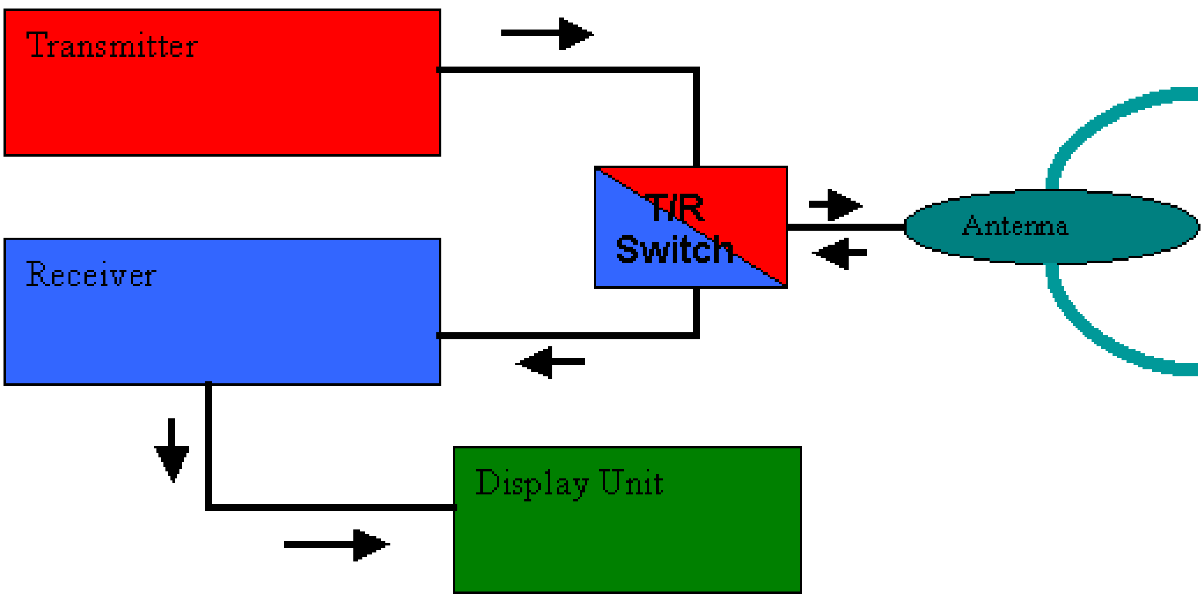


Figure 26: Radar Design

Radar output has two main forms which are velocity and reflectivity. Reflectivity measures how much precipitation is in an area. While velocity measures the speed and direction of precipitation in regard to the radar. Wave theory is strongly considered the physics of radar. In 1887 Heinrich Hertz discovered radio waves behavior. Showing invisible electromagnetic waves radiate acceptable electrical circuits travel the speed of light. These same properties are used to determine height of reflecting layer. Doppler Radar was discovered in 1842. Doppler is the theory sound waves change in pitch when approaching, but lower when going away. Doppler is used to calculate how the police cars are moving with sirens on based on the frequency of the siren. Doppler is also used in weather radar to find the speed of precipitation in the environment, usually use to detect how fast a storm or hurricane will be when it hits a certain area. Though very unpredictable doppler gives a better grasp on the depth of speed of precipitation in the atmosphere. With weather radar, it usually constitutes a map view of reflected particle in an area.

Depending on how drastic the precipitation is the color will vary in comparison to the precipitation on the map. With each color ranging from light to dark standing for a different level of reflected energy pulse precipitation. With the size and color depending on the precipitation particles and how intense these particles are in the area they are located. After taking these findings from the weather radar it can be predicted the approximate speed and rate at which it will rain at the ground.

Senor Array is a configuration of device which convert energy into electrical response. These electrical responses can be represented by sequence number and examined using digital signal processing algorithm. Sensor array are used in various devices from radar, sonar, wireless, and the list goes on. Sensor array is used so often due to the fact of its accuracy and its straightforward use.

In comparison to Radar, Camera, Lidar, and Sensor Array we decided to use Sensor Array for our design. We did not choose Radar due to the errors. When a major storm is about to hit many people turn to the news to see just how serious the storm is. Being a resident of Florida, many would notice how more the news is wrong more times than right in regard to predicting the level of a storm and even its direction. Radar images do not always accurately reflect everything in the atmosphere. If a radar is close enough to a coast and the beam of the radar is big enough it might reelect off of the sea returning a clutter. Also depending on the wavelengths some beams may not be reflected. Including the environment that the radar is in affects it also, being if there is a mountain, or building it can block the radars beam.

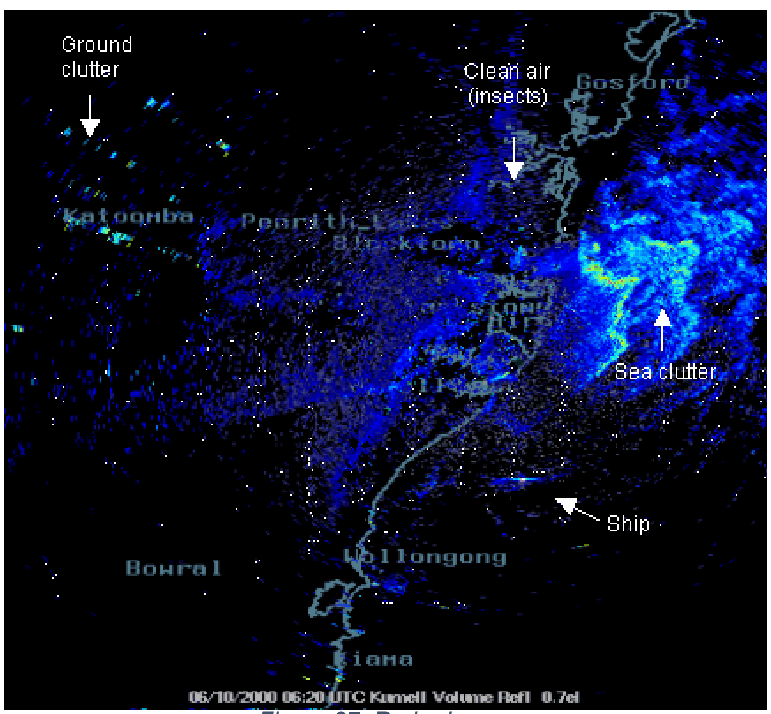


Figure 27: Radar Image

The return echo will become weaker the further away from the radar it is. Occurring due to the radar beam expanding with distance, allowing for more clutter and reducing the echo's intensity.

We also decided not to use camera. We chose not to use camera due to our project specifications and camera not having an in-depth enough output for our visual display system needs. Being that cameras work by using a lens which is a curved piece of glass or plastic to take beams of light bouncing off of an object and redirect it, so they come together to form an accurate image of what is present in front of the lens. It did not make the most sense when trying to create our design. Also, with the reflection of light and the slowing of waves from air to the lens makes it difficult to orchestrate our idea properly using camera.

Lastly, was Lidar which we decided not to do due either. Lidar is great being it is a Light detection and ranging instrument with very high accuracy and efficiency. Lidar being light pulsed lasers give precise three-dimensional information about Earths shape and characteristics of the surface. Being that we were only demonstrating physical objects and characteristics of the objects in front of our sensor the Lidar was more than needed. Including Lidar system usually consisting of laser, special GPS receiver, and a scanner. These seemed like more costly objects than using the sensors array we got online for a fraction of the price of these items. Lidar is also mainly utilized air mobiles, and to show information over broad areas. Which is something we did not need for our project. With the two types of Lidar, topographic and bathymetric. Topographic using near infrared laser to map land while and bathymetric usng water penetrating green light to measure water depths. Sensor Array became the obvious choice being that scientist and mapping professional mainly use Lidar to study manmade and natural landmarks, whereas we are designing a more personal visual display system for entertainment purposes. Sensor array was the best choice.

6.6 Possible Security Risks:

Due to the nature of our project, some security concerns come up. Concerns such as, will the images captured via the sensors be stored somewhere? Will the device be connected to the Internet or some type of network, and thus able to be infiltrated and the images taken? Luckily for us, our project mitigates these concerns because the images will not remain in memory.

The reason for this is that the actual microcontroller will not be able to store images due to its' limited memory space. As a result, this project represents a low security risk in terms of privacy. The sensors will not be connected to the network therefore no intrusions could be possible.

None of the components of the project are connected to the internet. Therefore, if anyone were to try and infiltrate our system, they would have to do so manually. Which is highly unlikely.

Another concern would be about privacy. Many people are not comfortable with having an image captured of them and then processed. However, these images are not stored anywhere and will not be able to be accessed by anyone. Furthermore, this project is simply a proof-of-concept project. Due to this, any privacy concerns can be mitigated by simply asking for consent to have your image taken. This will help mitigate any issues that this project will have and help alleviate any concerns people will have.

As important as the project itself is ensuring that the project is safe to use and is not prone to any attacks, infiltration, data breaches, etc. Therefore, it is of the utmost priority for us to ensure that there are no possible security breaches.

7. Prototyping:

It is important to ensure that each section of the project is working. Due to this we decided to begin prototyping each stage. The results below are from prototyping the main component which is the ultrasonic sensors. We did this as a proof of concept to ensure that we could stack our ultrasonic sensors together and run them, we decided to build a prototype connected to the Bela. The pictures below are pictures of the prototype connected to the Bela.

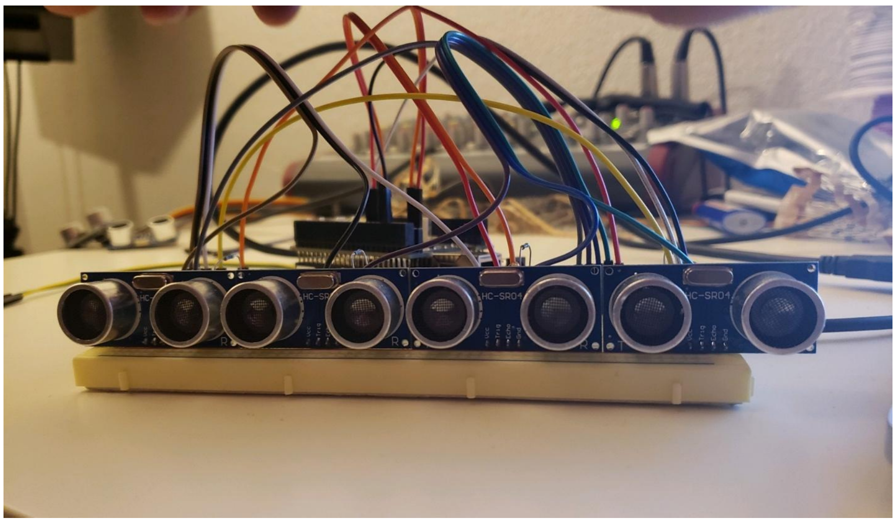


Figure 28: Prototype

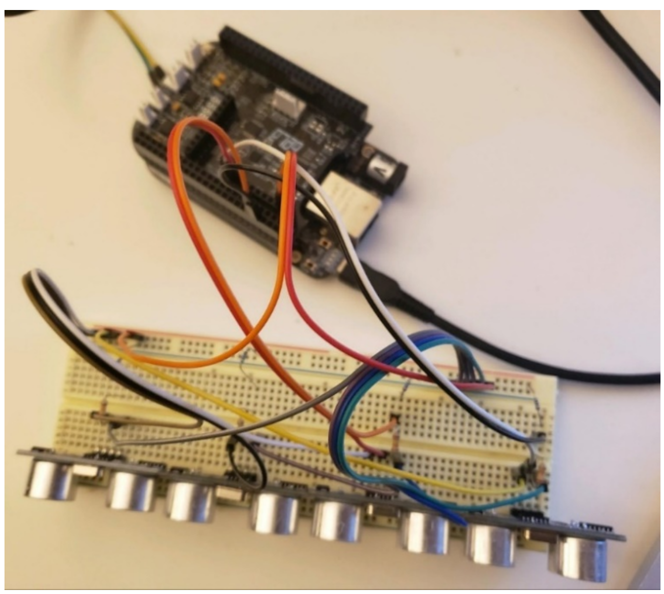


Figure 29: Prototype

The picture above shows the ultrasonic sensors connected to a breadboard aligned. We used this prototype to develop software in order to ensure that connecting ultrasonic sensors in an array in this fashion could work. We also connected the sensors to the Bela board to ensure that the Bela would not only be compatible with the ultrasonic sensors but we wanted to ensure the software on the Bela could take the ultrasonic sensors as input and use it to 2D Map and 3D Map. Furthermore, we tested the Ultrasonic sensors to ensure that they pick up objects located in front of it. Pictured below are graphs that indicate how close or far an object is from the sensor along with the pulse-widths and the actual object in front of the sensor.

The first set of images represents the sensor input with no object placed in front of it, the second set of images represents the object in front of the sensor, and the third set of images represents the object very close to the sensor.

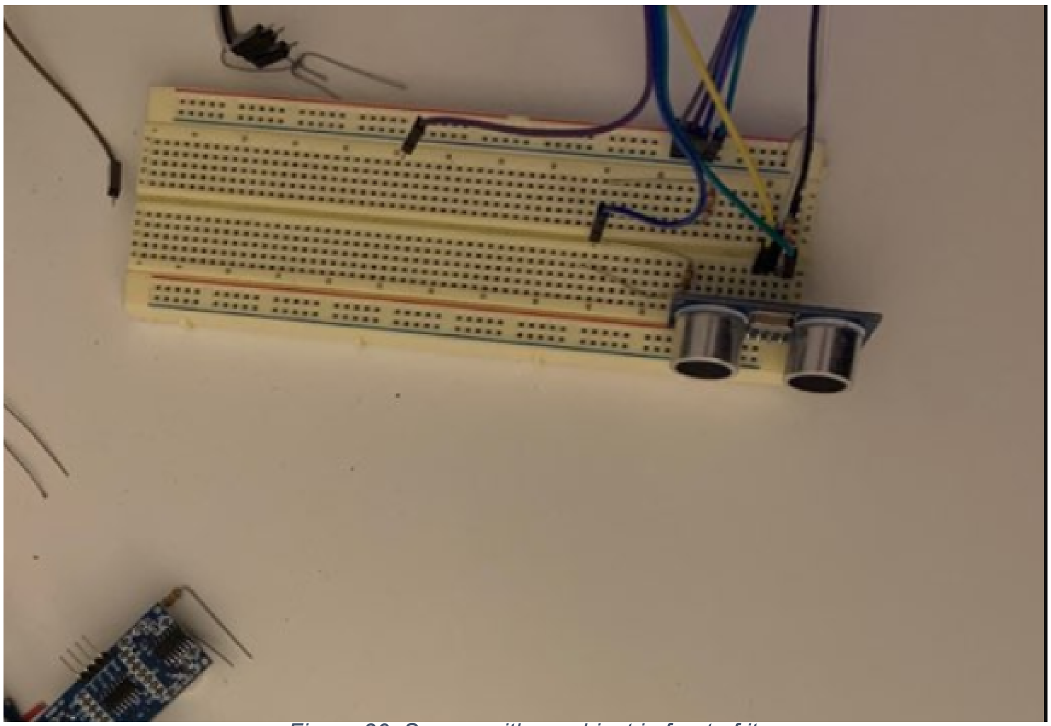
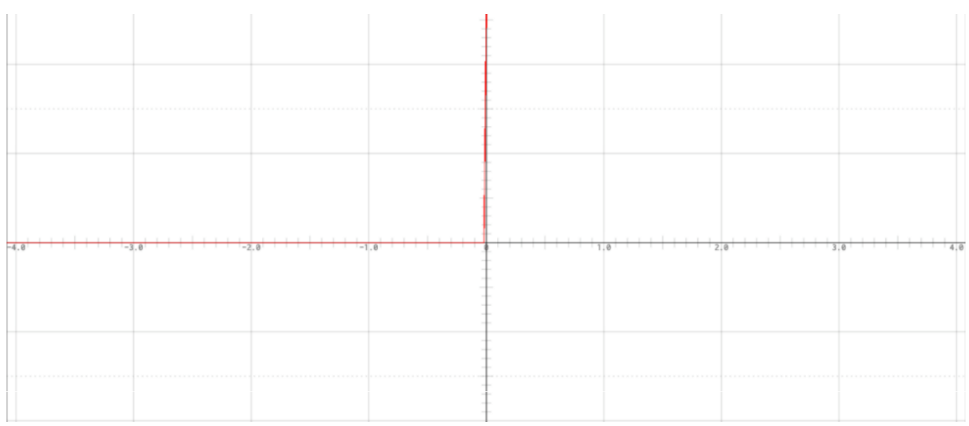


Figure 30: Sensor with no object in front of it



In the graph above, the red line represents the time of flight, which is the time a pulse is sent to an object and the returned to the sensor. It is exactly at 0 because it sends a pulse out but because there is no object in front of it, the sensor does not pick back up a pulse. This is how the pulse-length is calculated. Furthermore, the distance is calculated by the duration of the pulse multiplied by the speed of sound divided by 60.

```
bela: pulseLength: 262, distance: 102.431778cm
bela: pulseLength: 262, distance: 102.431778cm
bela: pulseLength: 259, distance: 101.258896cm
bela: pulseLength: 260, distance: 101.649849cm
bela: pulseLength: 262, distance: 102.431778cm
bela: pulseLength: 263, distance: 102.822739cm
```

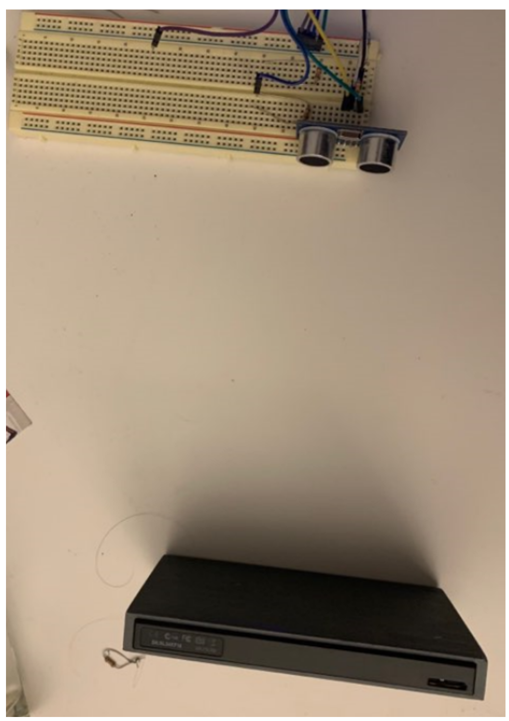


Figure 33: Sensor with object in front of it

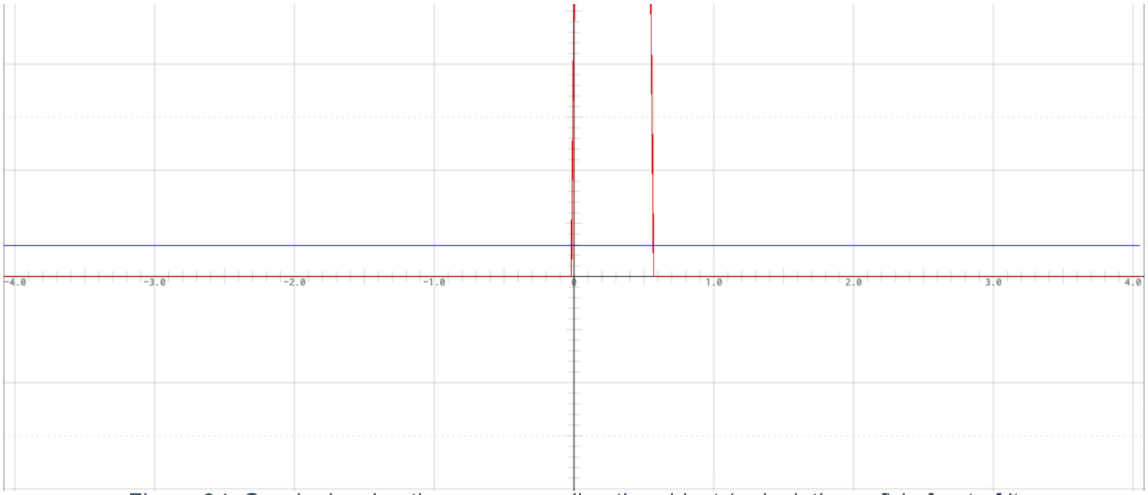


Figure 34: Graph showing the sensor reading the object (or lack thereof) in front of it

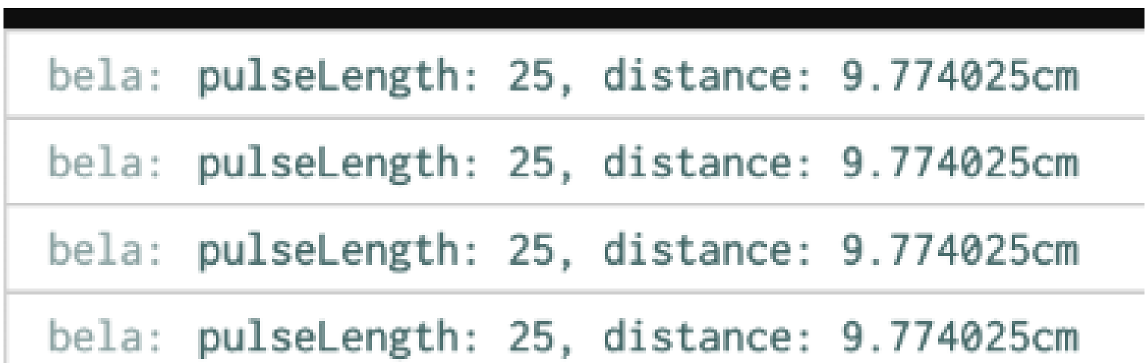


Figure 35: Pulse Length and Distance

In the previous set of images, now that there is an object placed in front of the sensor, the sensor is receiving a pulse back. This is represented in the second red line that appears. The distance between these two lines represents the time of flight, and that number divided by two represents the actual pulse width.

The blue line represents the distance in centimeters. The graph is scaled. Furthermore, it is shown now that the distance value has decreased. This shows that we can now detect distance levels via our ultrasonic sensors. This is good because it shows that we are now able to take in accurate data from our ultrasonic sensors. We can now use this data to control our outputs. This essentially proves our proof-of-concept, that we can take in information and extrapolate data from it. Since it is proven we can now do that now the challenge becomes making sure we can correspond data from the inputs to the LED Cube as an output.

The goal would now be to ramp it up and add more ultrasonic sensors. It would allow the sensors to pick up further data points, the goal would be to take those data points and then use the Bela to map those data points to a rendered image.

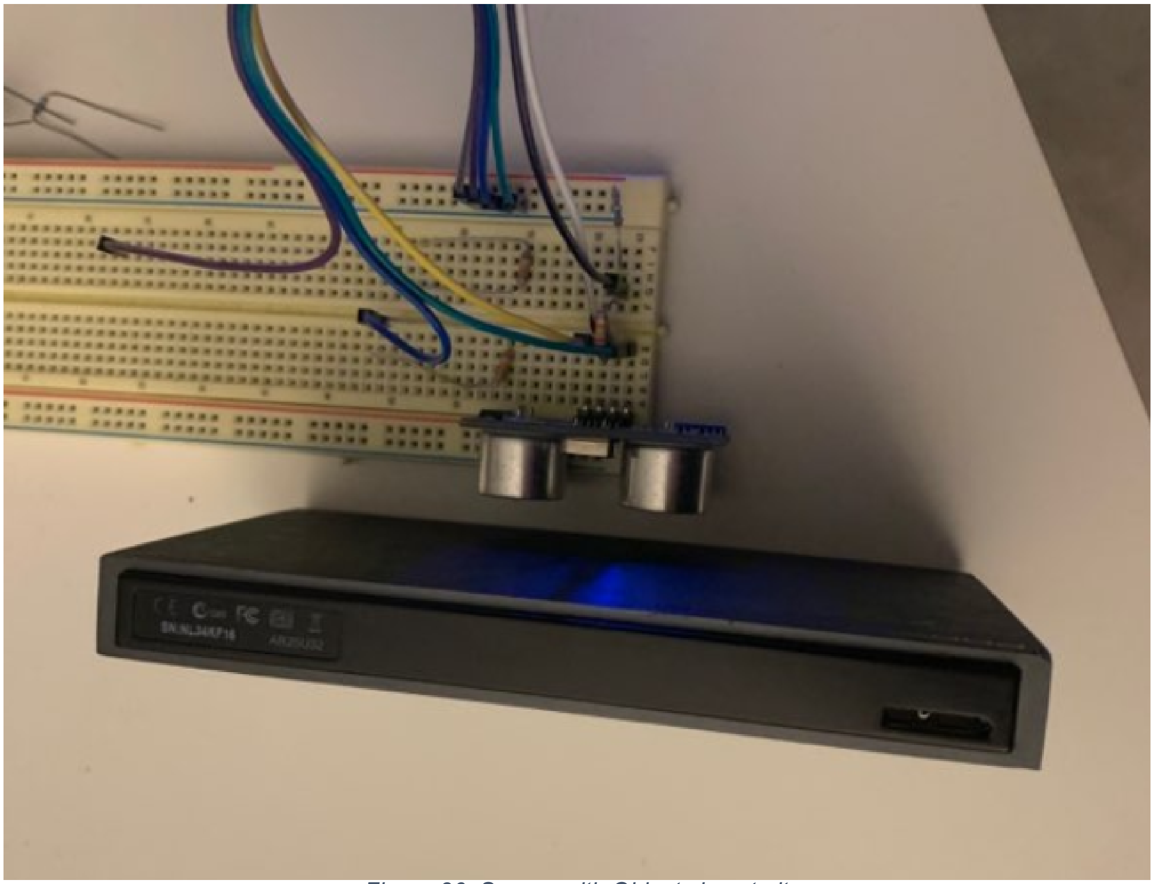


Figure 36: Sensor with Object close to it

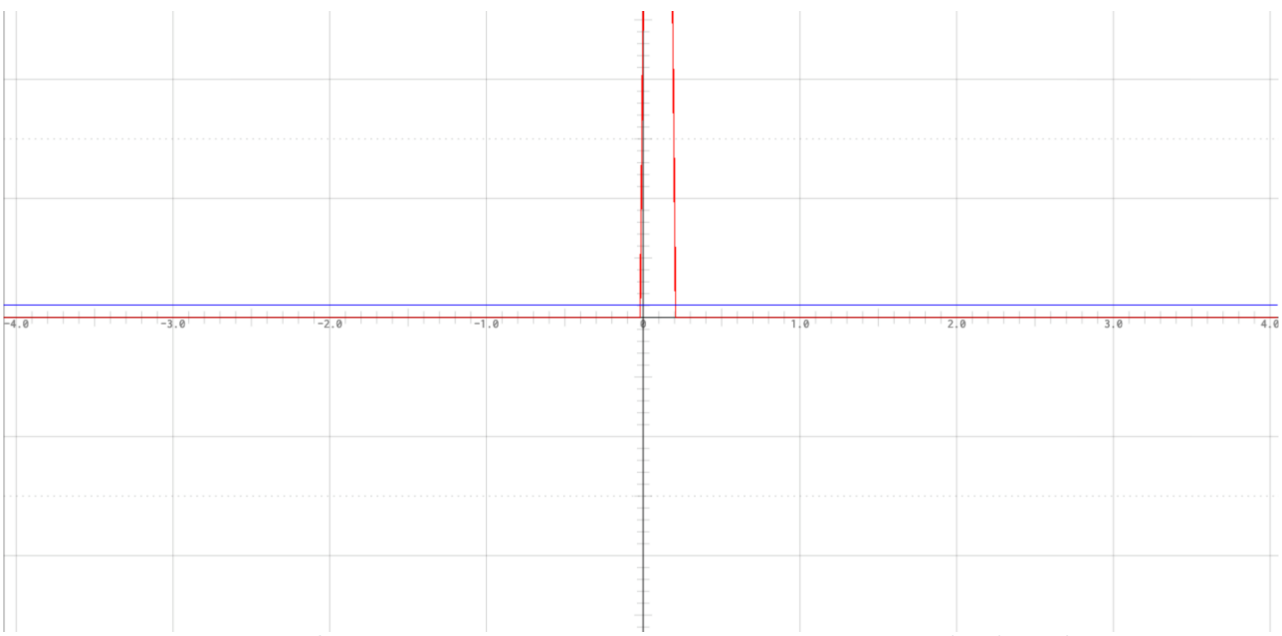


Figure 37: Graph showing the sensor reading the object (or lack thereof) in front of it

```
bela:  pulseLength: 9, distance: 3.518649cm
bela:  pulseLength: 9, distance: 3.518649cm
bela:  pulseLength: 9, distance: 3.518649cm
```

Figure 38: Pulse Length and Distance

The last set of images show what happens when an object is close to the sensor. The two red lines are closer together and the blue line is closer to 0. This shows that the sensor is sending and receiving the pulse in a short amount of time.

This is a proof-of-concept for the ultrasonic sensors. Now, the objective will be to scale it up and place more ultrasonic sensors. This will allow the sensors to pick up more data points, the goal will be to take those data points, and then use the Bela to map those data points into a rendered image.

The main risk of using more ultrasonic sensors is the amount of crosstalk that will be received however, we should be able to reduce the amount by de-muxing and by carefully selecting which sensor is on or not. This is more explained in the following section.

Pictured below is the schematic for the prototype that we built. Each sensor is connected the de-mux which is then connected to the control. The control switches which de-mux is currently operating, then the de-mux controls which sensor is on at any given moment. This will allow us to control which sensor is on. Not only does this reduce the amount of software that needs to be made, but this allows us to help reduce crosstalk between sensors. These two benefits will greatly improve the efficiency of our project.

The image below is the schematic of the scaled-up version of this prototype with all the 256 sensors. This amount of scaling requires a sturdy frame in order to make sure all the sensors are properly in place and not moving around. To this end, we will use L-brackets to ensure that they remain in place.

Our design framework is essentially a collection of modular components that can be reused, bound together by a style guide and detailed documentation. A good design framework is fundamentally restrictive but versatile in implementation. Teams using design systems have more flexibility to concentrate on web development and user travel performance because they do not design interfaces and visual solutions from scratch continuously.

Furthermore, we were able to make our schematic such that if we ever wanted to scale it up, we could. This allows for easy scalability and in the case our project ever gets sponsored, we will be able to use the work we have done here to create bigger and better designs.

When it comes to scaling design, it is important that the design is originally made in such a way that can work regardless if there is one component or a hundred. Therefore, with this process in mind, we optimized our design such that when we do eventually scale it up, the design will retain its efficiency while also allowing us to do more with the project.

These are important concepts that needed to be maintained while continuing the course of our project and due to keeping these frameworks in mind, we were able to make a project that works at any scale.

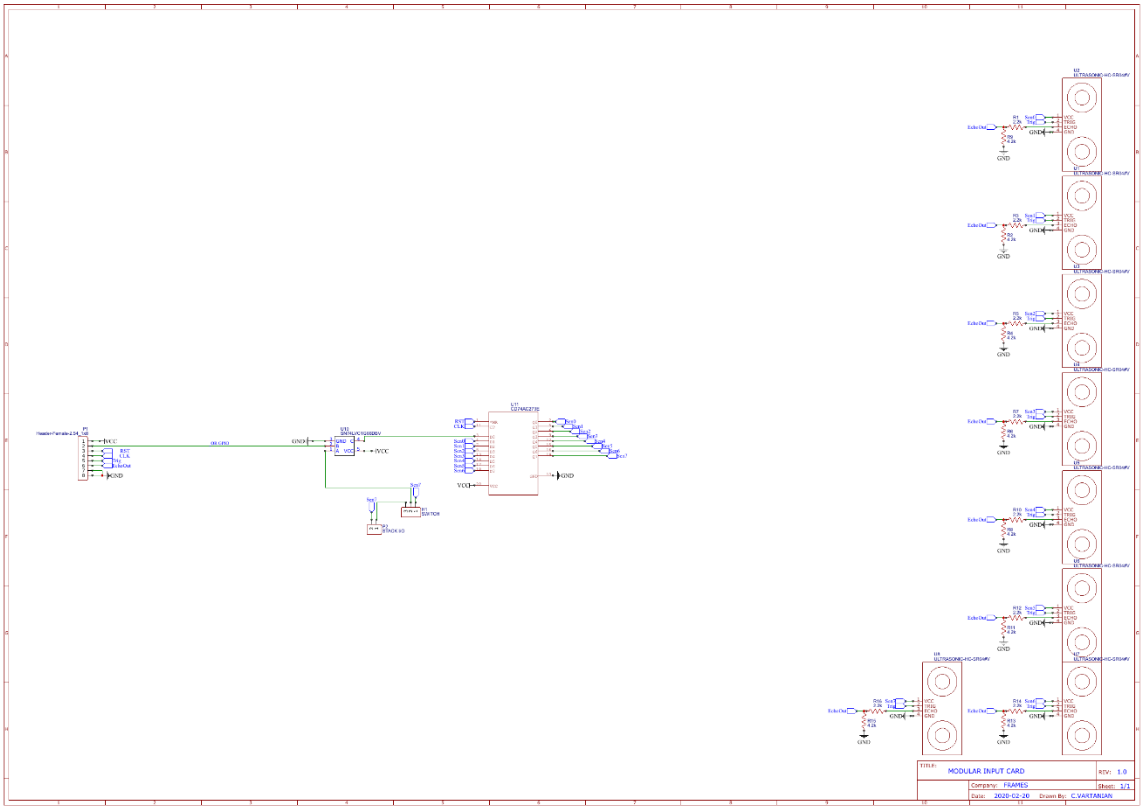


Figure 39: Schematic

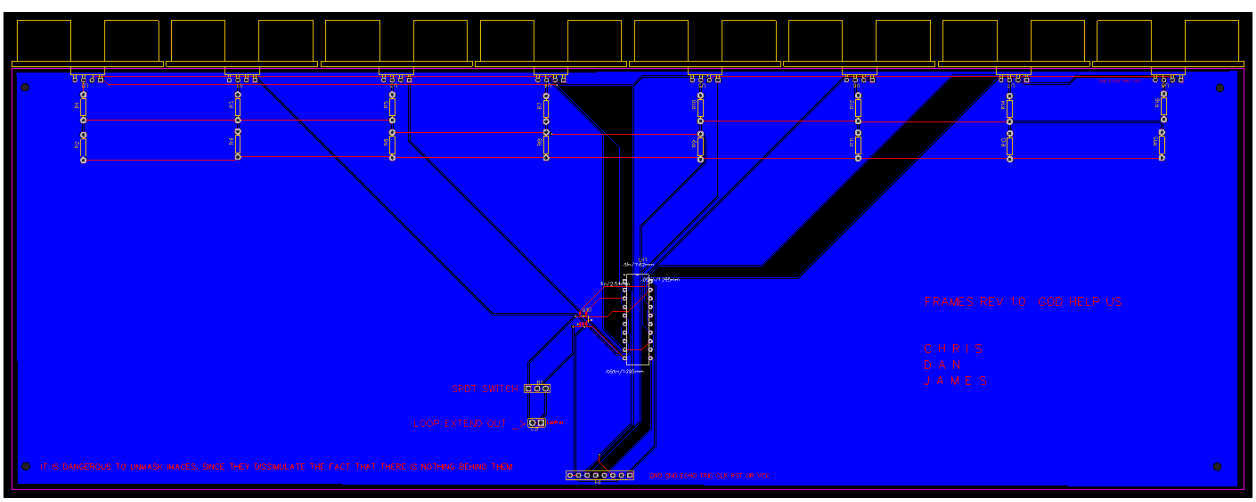


Figure 40: Schematic

The images above show a full schematic build of our PCB board while the image below shows the physical PCB board with the ultrasonic sensors connected.

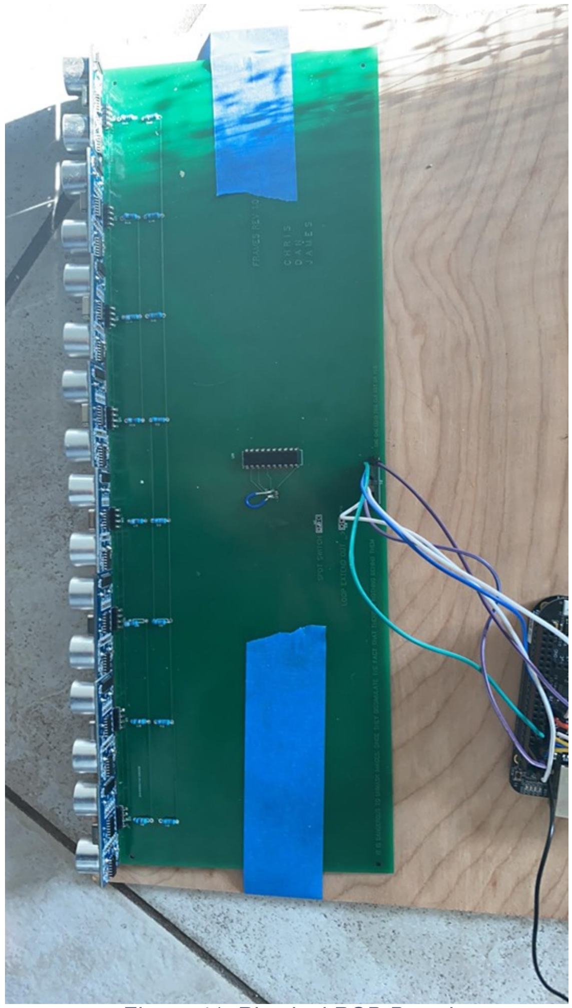


Figure 41: Physical PCB Board

The image above shows our PCB fully constructed and build with ultrasonic sensors and connected to the Bela board. We were able to get 20 printed in the case that one or two of the PCBs did not work and in the end it served us well because 3 of our PCB boards did not work.

Furthermore, we will prototype the LED Cube by building a simple 4x4x4 cube to ensure that the LED Cube can be extended to a 16x16x16 cube. Having a small prototype for the Cube, ensures that we can test it and ensure its functionality before we expand on it and build on it. Once we see if it works on a small scale, it should be easily expandable. We will build the frame, then connect the LEDs together, then solder it all together. Afterwards we will connect the prototype ultrasonic sensors, the Bela board, and then the LED cube and test out the entire project on a small scale. Once the entire project is tested on a small scale, we should be able to scale it up to more properly fit our project requirements.
Pictured below is what our 4x4x4 prototype will look like. We will use a guide posted on the internet to follow in order to construct our prototype (and subsequently or LED Cube) with the explicit permission of the owner of the document (Noted in the Appendix)

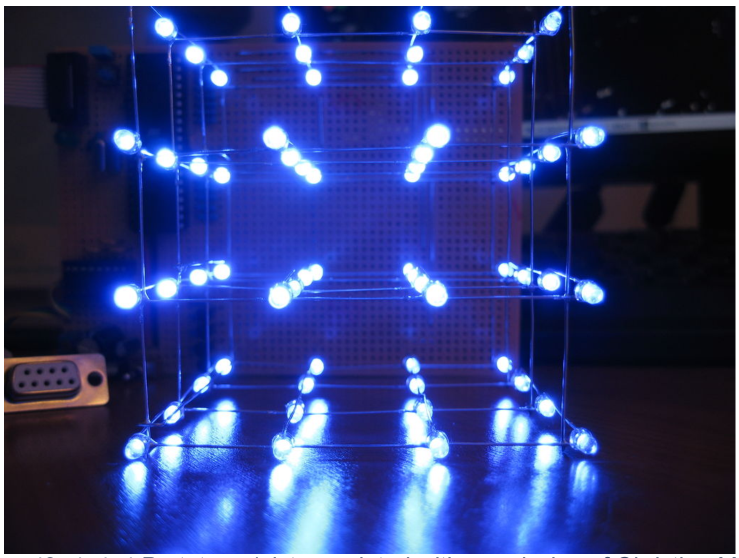


Figure 42: 4x4x4 Prototype (picture printed with permission of Christian Moen)

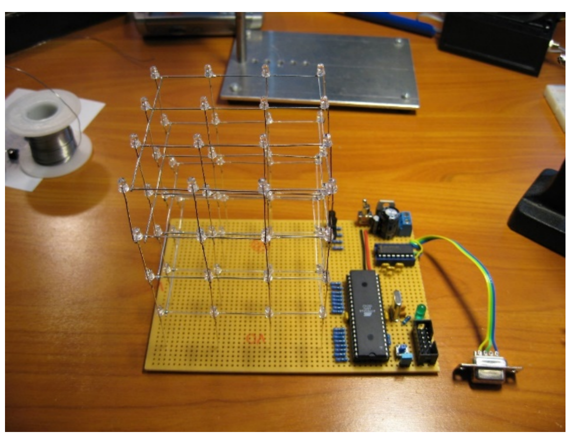


Figure 43: 4x4x4 Prototype (picture printed with permission of Christian Moen)

The above images represent a smaller scale version of the Cube we wish to build. This cube is a 4x4x4 design whereas ours will be a 16x16x16 design.

8. 3 to 8 Decoder:

Simply utilizing the Bela's 8 available analog pins as inputs for the sensor array would not be enough to directly sample the entire array at once. If we are to opt for a progressive scan technique wherein we utilize some of the output pins as "selects" to parse through the sensor groups (literally turning each group on and off), we will need to utilize a decoder. As the array is intended to be 16x16 and we have 8 available inputs we need to section the array into a minimum of 32 groups. As such we would need 5 available pins to describe 32 groups if we are to use a binary system of turning each group on and off.

Texas Instruments' CD74HC238 is a 3 to 8 decoder that can be daisy chained with 4 more units to form a 5 to 32 decoder. As each sensor on the array takes in a 5v Vcc line as a power signal, a fairly obvious wiring set up occurs. By connecting the power rail of each of the 32 groups to one output of the decoder and giving the decoder a 5V VCC signal. We will be able to select which group is on and active sampling by simply sending data to the 3 input pins on the decoder.

The CD74HC238 is specifically a cheap and readily available 3 to 8 decoder whose power consumption is relatively agnostic (only 80-µA!). It functions as a standard 3 to 8 decoder while using whatever signal is sent to the VCC input as an output signal. Wonderfully it will respond to both 3.3V and 5V inputs. This will allow us to connect either the digital out, or the analog outs to the sensor array, whichever is more readily available in the final design.

As we are using the decoder to directly power the sensor array, we will need to directly wire each of the 32 outputs directly to the VCC input of the sensor array. As such 8 wires will be connected to each output to turn on each group. Using this logic, we have opted for the 238 version of the CD74 as this version is active high. After distributing power to each of the VCC inputs we shall wire the signal pins from each of the modules directly into the Bela's analog inputs. As such, there will be 32 wires going into each of the 8 available inputs but only 1 of these 32 wires per input will be hot as they will be grouped and selected by the select logic from the decoders.

The same logic that allows us to take in the input of our sensors is going to be used to output to the LED cube. The Bela comes equipped with 16 Digital I/O ports that we will section off into 3 sections of 4 pins each. Each of these sections represents one of the dimensions (X,Y,Z) on the LED Cube. At 4 pins per section we can address 16 separate coordinates per dimension, the exact amount required to address every LED individually. Of course, 12 output pins alone are unable to direct map to each of the 4096 LEDs so we will have to again utilize the CD74HC238 decoders, in this case arranging them as three separate 4 to 16 decoders.

8.1 Truth Table of Output and Input Decoder Schemes:

Truth Tables are tools used in Digital Design that allows the designer to map out specific outputs for a corresponding input as the result of a Boolean expression. A Boolean Expression is an expression that can be either true or false, 1, or 0. A truth table is a visual tool, in the form of a diagram with rows & columns, that indicates the reality or falsity of a compound premise. It is a manner of organizing information to list out all viable situations from the furnished premises. Truth tables are sleek, useful logic-tracking diagrams that exist not just in mathematics but also in computer science, electrical engineering, and philosophy. The terminology can differ based on which business you are working in, but the basic principles remain the same. It is a flexible, interdisciplinary device.

The Truth Tables demonstrate the necessary logic set up for both the Digital Output to LED Cube and the Power Select system coming from the Bela and toward the Frame Sensor array. Please see the Appendix for the Input and Output Truth Tables. They are detrimental to the functionality of the entire project. Therefore, it is important that they are accurate and to make sure they are accurate we will test each input and output to ensure that everything is functioning properly. These truth tables will allow us to properly output each LED to the proper ultrasonic sensor. It is imperative that this logic is correct otherwise our LED cube will not properly output the input the sensor array is receiving. Each of the inputs will be mapped to a specific output which will allow us to do a 1:1 mapping.

Crosstalk occurs when two ultrasonic sensors receive each other's signal. Due to the number of ultrasonic sensors we have next to each other, this was an issue we had to face. Crosstalk could lead to faulty data and faulty data will provide the wrong output. Thus, we needed to arrange our sensors and program them in such a way so we could eliminate this issue as much as possible.

The way many ultrasonic sensors combat this is by introducing a concept called multiplexing. Multiplexing allows each burst from each sensor to be delayed such that no two sensors send a receive a signal at the same time. Due to each burst being sent at a different time, the possibility of any sensor mistaking its neighbor's signal as its own is reduced. Many ultrasonic sensors incorporate the ability to multiplex that is activated by wiring the inputs on every sensor to a remote connection. However, ours did not therefore to remedy this situation we basically included an OR gate in our PCB design that will turn on and off whichever ultrasonic sensor is not being used. The OR function basically states that an output is true if A OR B is true. In our application, this basically means that the sensors will not turn on unless it detects a target, in which case it'll see which sensor the target is in front of and then proceed to turn off the other sensors. An image of

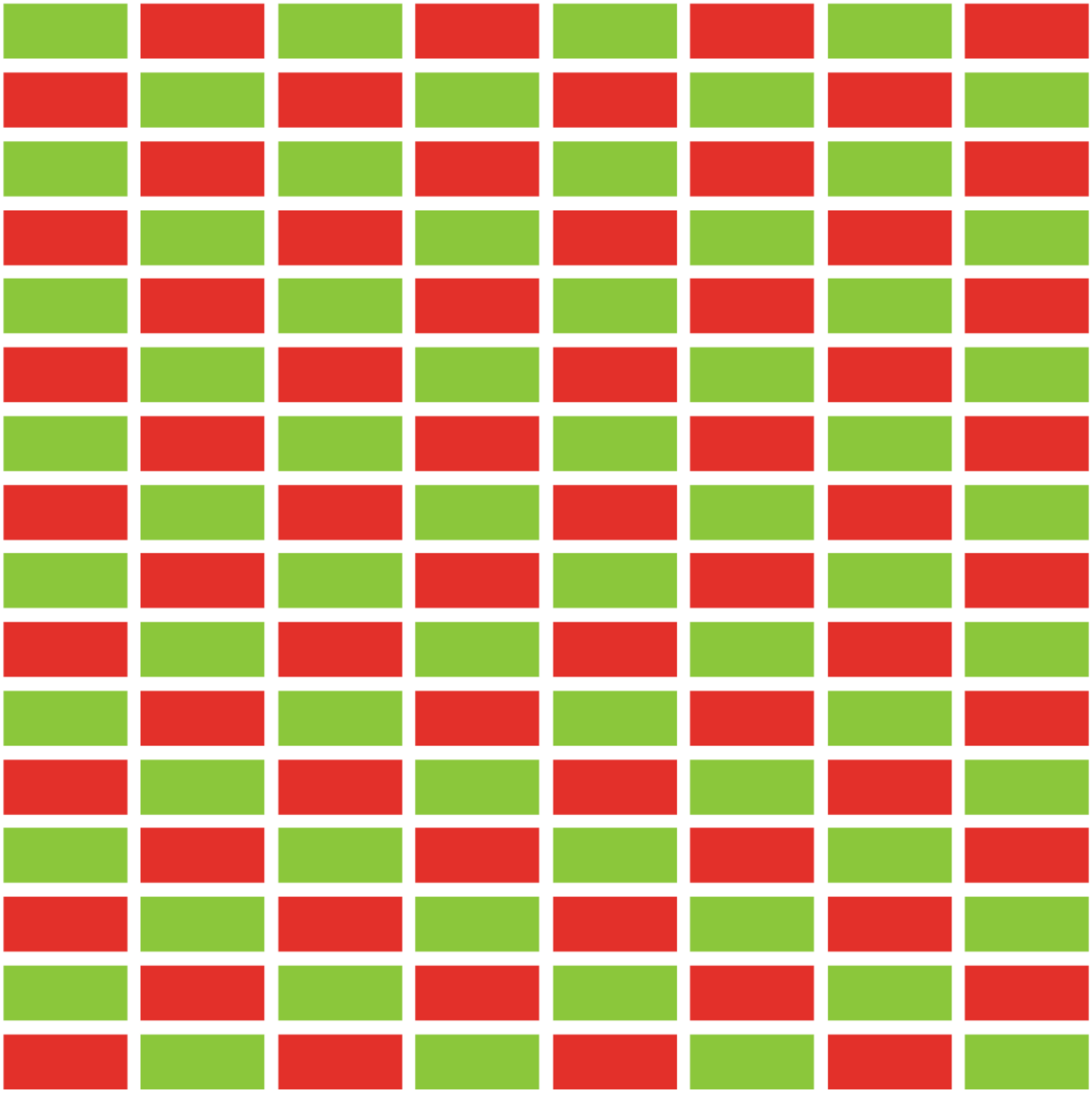


Figure 44: Multiplexing LED Cube Outputs

A proposed arrangement of sensors for sampling groups that could negate the effects of crosstalk, the green group is sampled and then the red, thus defeating any tendency for one module Echo transducer to pick up the pulse from another module's Trig transducer. By doing this, we will increase the efficiency of the sensors. This arrangement will prove beneficial to us because it will increase the quality of the image that will be displayed.

8.2 Sound Component

Not only does the Bela function as an amazing platform for the development of real time analog processing tools, but it also is able to output real time audio at high quality sample rates. A stretch goal of our project was to utilize this feature such that we would be able to have a sound output that followed our visual outputs. We opted for a simple execution, but this methodology could have been scaled for a much more sophisticated audio output quite easily.
By casting our distance value as a floating-point variable in our system we can practically tie the variations of the sensor to any output we want with relative ease. Understanding this, we crafted a simple sine wave generator whose amplitude was inversely controlled by the distance and whose pitch was modified by a flat amount. The sinewave wrote to a buffer in the audio thread that is sent out to a line driver and (in our case) was connected to a Fender Guitar Cabinet and Amplifier. By inverting the relationship between distance and amplitude, an uncanny valley was created in which the speaker appeared to get quieter and eventually completely silent as it was approached. This effect could have been better achieved through a logarithmic relationship between distance and amplitude but was enough in its linear relation to create an eerie effect.
In short, in achieving this stretch goal we were able to add a whole new dimension to our system and further contextualize how we experience the spaces around us. The unsettling factor of the inverted speaker converts the user with a non-logical relationship to the space they find themselves in. In forcing the user to reconsider their understood notions of space, and how sound propagates, our mission of recontextualizing space is further reinforced.
While we could have simply used a multi pin approach to individually activate and deactivating each sensor, it made more sense to opt for an analog sequencer to simply handle the scanning of the sensors. By using a multi stage D-Flip Flop and connecting the output lines to each sensor and also to the data line of the next Flip Flop we crafted a kind of sequencer that was able to successfully pass a power signal to each sensor individually. The primary bonus of utilizing this schema came in the form of a severely reduced wire count. Instead of manually activating each sensor we could now simply use one pin to clock the sequencer and scan through every single sensor with ease. Sensor Locality was preserved by keeping track of each clock pulse and understanding which sensor started (we know we start on sensor 0 so at clock pulse 32 we will be on sensor 32). This drastically simplified the project as we no longer needed to worry about an absurd amount of wires to carry VCC lines and could focus our attention on syncing the system via more manageable software setups.

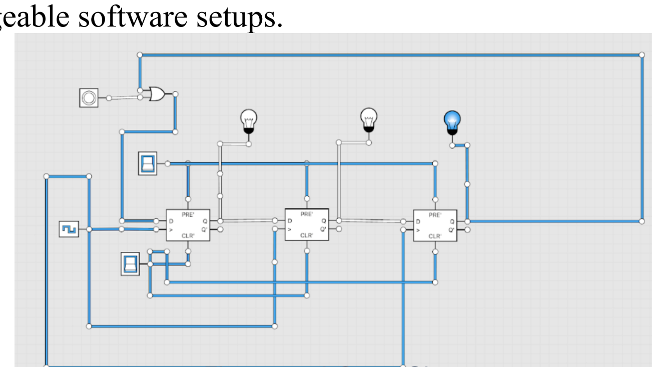


Figure 45: 3 Stage Sim

A 3 stage simulation of our flip flop sequencer, Light Bulbs represent the VCC inputs on each sensor

9 Frame Structural Design:

As previously stated, the input of the Frames system is a cartesian array of ultrasonic rangefinders. In order to properly power and support the large number of sensors a system of rails and brackets must be erected. Each Ultrasonic module (HC-SR04) has four mounting holes in the corners of the backing PCB. By making a series of rails with equally spaced holes we will be able to properly brace the frame against the weight of the sensors. Additionally, the spacing of the pins that come preinstalled on the HC-SR04 require a spacing of about .5 cm above and below each module. Thankfully, this spacing allows us to square off the arrangement of every single sensor and fit wiring connections at the pins just between each of the sensor modules.

From a wiring perspective, the HC-SR04 requires one contact for a 5V VCC line, one contact for a trigger in, one for an echo out, and one for ground. Obviously, we cannot consolidate the signal lines of all the sensors as this would fuzz all the data. This aside we can consolidate the ground and power lines so that they are out of the way and not coming into the microcomputer system as over 500 wires carrying ground and power. We can directly install a tap from out power supply into the frame that carries both ground and power lines. From here there are two apparent methodologies for distributing these lines. We can simply take wires from the VCC and ground tap and individually connect them to the various sensors in the Frame. This is an easy solution but still comes with quite a lot of wire bloat (which is best to be avoided if one wants to keep their sanity). A potentially more difficult to design solution, but one that comes with far less wire bloat is to use electrically connected rails as ground and power lines.

Using the actual mounting rail as a ground line is a common practice in the realm of modular audio products. Most mounting rails are made from metal and are directly screwed into a faceplate or PCB and thus, creates a solid structurally sound method for grounding. This is generally referred to as “Chassis Grounding”. Not only can we utilize the mounting rails as a grounding system, but we can also erect rails to connect the 5V VCC line off.

The rail-based power and ground system, while more convenient in terms of removing wire clutter and general management (having continuous ground and power rails would allow easy hot swapping), comes with caveats of added design complexity and potentially more expensive materials cost. Since the mounting rails in this case would become part of the greater circuit they obviously need to be conductive. This removes the possibility of using plastic rails or brackets which, while potentially cheaper to produce, would be unable to carry any ground or power signals. Metal mounting rails are widely available for a decent price but it does always pay to consider the financial cost of opting for such a potentially expensive option.

9.1 Power Supply:

For the visual display system, a power supply is needed to power the LED’s for the 3D imaging. Most visual design system designers typically tend to choose wall power. Wall power is the common choice due to being widely convenient and only needing a detachable electric plug to use. With most buildings in America hoisting multiple wall power outlets. Another reason being most wall power outlets provides 15 to 20 amps of electrical current, which is about 1800 watts in retrospect. This number of watts is enough to power a fully occupied house with several appliances. This makes wall power very suitable in most circumstances where almost any electrical dependent object with a power outlet cord can be used.

Another key power supply is the Universal Serial Bus or for short the USB power supply. The Universal Serial Bus is a protocol connecting peripherals to a computer through a port. The Universal Serial Port features standardized ports designed to work with many various types of hardware device. According to "" Most modern devices such as digital cameras, printers, scanners, flash drives, cell phones, iPods and other MP3 players use some variation of the USB port in their design.

According to "The first Universal Serial Bus technology began development in 1994, co-invented by Ajay Bhatt of Intel and the USB-IF (USB Implementers Forum, Inc). The organization is comprised of industry leaders like Intel, Microsoft, Compaq, LSI, Apple and Hewlett-Packard."

By Supporting and adopting comprehensive specifications for all of the aspects of Universal Serial Bus technology.

Before the use of Universal Serial Bus computers would use serial and parallel ports as plugs to connect devices with computer and to transfer any data necessary. The individual ports would be used as peripherals for various hardware like computer mouse, printer, external keyboards, and even gaming remotes. Having expansion cards including custom drivers we would have to connect devices. The Parallel ports transfer data at a speed of about 100 kilobytes every second, but serial ports range for transferring data at 115 to a little over 450 kilobits every second. With some ports not being able to run at the same time. With times changing a high volume of incompatibility between devices and computers made way for a demand for Universal Serial Bus. The Universal Serial Port worked great due to being a sole connection for port types. Universal Serial Bus made it so there was instant connection between devices and host computer without needing to disconnect or restart the computer being used making the Universal Serial Bus highly efficient. Just one USB port can handle over 100 devices, offering a collective compatibility for each device. With the Universal Serial Bus 1.0 debuting in1995 and transferring data at a rate of 12 megabits per second.

There was a revision version done for the USB 1.0 called USB 1.1 which transferred data at a full speed rate of 12 megabits every second and also ran at lower speeds of 1.5 megabits every second for low bandwidth device. Since USB 1.1 was more efficient, USB 1.1 ended up being used more by consumers than the previous USB 1.0.

In the year 1998 iMac G3 ended up being the first consumer computer to discontinue ports known as legacy ports which were serial and parallel to change them for Universal Serial Bus. With this change it helped to make way for a market strictly for Universal Serial Bus peripherals and using there ports rather than other ports for devices. The Universal Serial Bus was easy to use, self-powered from the computer, and technical aspects allowed it to be better than over port options.

With Universal Serial Bus having a transfer forty times faster and 480 megabits every second, the introduction of Universal Serial Bus 2.0 debuted in the year 2000 and became an official standard the next year in the computer industry. The Universal Serial Bus 2.0 had a high-speed transfer rate aside from this the Universal Serial Bus was able to operate at slower speeds for devices to function at the same level. The Universal Serial Bus 2.0 also offered plug and play capabilities for the various multimedia and storage devices used. This new feature offered more user features that did not exist in Universal Serial port 1.0 or 1.1.

Universal Serial Bus 2.0 adds support for power sources with USB connectors, multiple interfaces, and capability for two devices to interact without needing a different USB host. With the year 2000 introducing many technological advances it also introduces the Universal Serial Bus flash drive, which is a read write storage device similar to a floppy disk first sold by IBM and Trek Technology. The Universal Serial Bus flash drive could at first hold up to 8 megabytes of information. Currently more than two decades later, the storage capabilities have gone upwards of 256 gigabytes for just one flash drive. than a decade later, storage capacities have surpassed 256 gigabytes for a single drive. With the latest version being Universal Serial Bus 3.0 with a data transfer rate up to 4.8 gigabits per second. The Universal Serial Bus 3.0 provides backward compatibility with the prior USB 2.0 device and ports. The Universal Serial Bus 3.0 cable and device can be plugged into a prior USB 2.0 port but won't have the same 4.8 gigabytes transfer rate. The Universal Serial Bus 3.0 technology is able to upload and download simultaneously from separate wire lanes of traffic enhancing speed of reading and writing data. Allowing devices to charge and function faster while using energy more efficiently. Universal Serial Bus 3.0 also allows devices like high resolution cameras and audio and external drives that are optimized at high bandwidths to operate more effectively. With Universal Serial Bus 3.0 products becoming available in 2010, the Universal Serial Bus expanded rapidly in a short period of time with full implementation of super speed still in the works. With a delay in the Universal Serial Bus 3.0 it made it difficult for Universal Serial Bus to compete with other power supply due to the low output of watts. Universal Serial Bus has a high bandwidth radio technology that transmits data from devices that are connected to it like printers, cameras, flash drives and even streaming video at fast connection without the use of cords or cables. The wireless Universal Serial Bus operates best within a short distance being able to transfer data at about 110 megabits every second at 30 feet and 480 megabits every second from a 10 feet distance. With wireless Universal Serial Bus working to expand user options for mobile computing while maintaining connectivity for a number of standard devices. Since Universal Serial Bus output, a low voltage in comparison to wall power or batteries, it is usually used with smaller devices. USB power supply can only draw a max of 500 milliamperes from computers and are commonly used with breadboards, printers, and components that only need a minimal number of volts to function.

The last key power supply option is a battery. Batteries were invented by Alessandro Volta in 1800. Batteries are around us everyday bus many people do not seem them often due to being inside of many objects. Battery are essentially a device that stores chemical energy that is then converted into electricity. Batteries are small chemical reactors, which has the reaction producing energetic electrons, that flow through the external device when needed. Dating back to 1800 batteries have been around for a very long time. In the year 1938 the Director of the Baghdad Museum found now referred to as "Baghdad Battery" in the basement of the Baghdad museum. When the Baghdad Battery was analyzed it showed that the battery was dated to 250BC with a Mesopotamian origin. Much controversy has surrounded the earliest battery suggesting electroplating which is pain relief in many cultures. Benjamin Franklin an American scientist and inventor used the term battery in 1749 when Benjamin was doing an experiment with electricity using a set of linked capacitors.

The very first true battery was invented by Italian physicist Alessandro Volta in 1800. The lead-acid battery was later invented in 1859 and is still used till this day to start most internal combustion engine cars today. Being the oldest example of a rechargeable battery. Today batteries come in various sizes from large Megawatt which store power from solar farms or substations to guarantee stable supply in entire cities down to tiny batteries used in electronic watches.

According to "Batteries are based on different chemistries, which generate basic cell voltages typically in the 1.0 to 3.6 V range. The stacking of the cells in series increases the voltage, while their connection in parallel enhances the supply of current. This principle is used to add up to the required voltages and currents, all the way to the Megawatt sizes."

The first battery was known as the voltaic pile. The voltaic pile consisted of copper and zinc disc which were piled on one another separated by a brine-soaked cloth or cardboard. Batteries are a great source of power for many reasons, but the main being how movable they are. Batteries are in cars, laptops, and even airplanes. Which oddly enough are all objects that either move or can be moved. Battery can be designed to hold a large amount of electricity to power any specific object and is a great power source for any object that is mobile.

For our design after considering the pros and cons. We decided to use wall power. Universal Serial Bus was highly convenient and efficient for many devices, but with our visual display system the USB would not be powerful enough to power it to function. For wall power we will only have to attach a power cord. Harvey Hubbell II patented the first detachable electric plug in the United States.

According to "Hubbell's detachable electric plug revolutionized the way electrical wires were connected or disconnected from a power supply. At the time post terminals would extend out from a wall and any type of electrical device had to be hardwired to the power source. This was extremely inconvenient—and dangerous."

So, if a device is moved without wires connected to the right polarity a very dangerous short circuit may occur. Also, the wall power will give us enough energy to power the visual display system. The USB power supply was too weak to power all of the LEDs for imaging. For battery, due to the design constraints, we were unable to find a battery powerful and compact enough to work with our design.

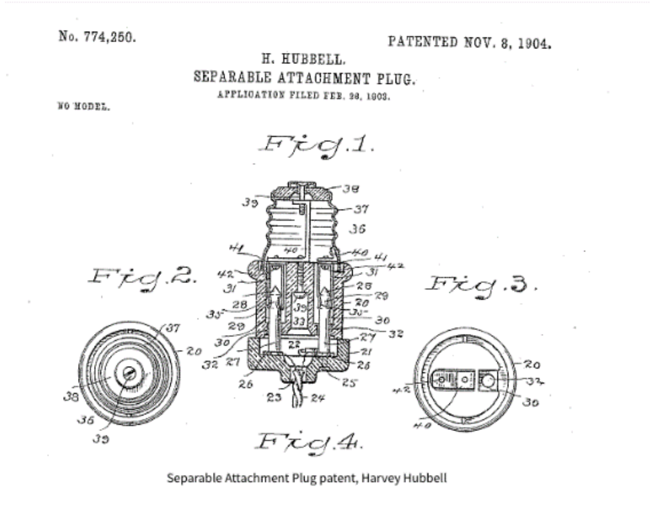


Figure 46: Patent

As a result of all these design constraints, plus the conveniences of wall power, we have decided to go with that route to power our LED Cube as well as many other aspects of our project. Below is an image of our finished project with the power supply (9v battery) as well as a USB connection to the Bela to power the Bela.

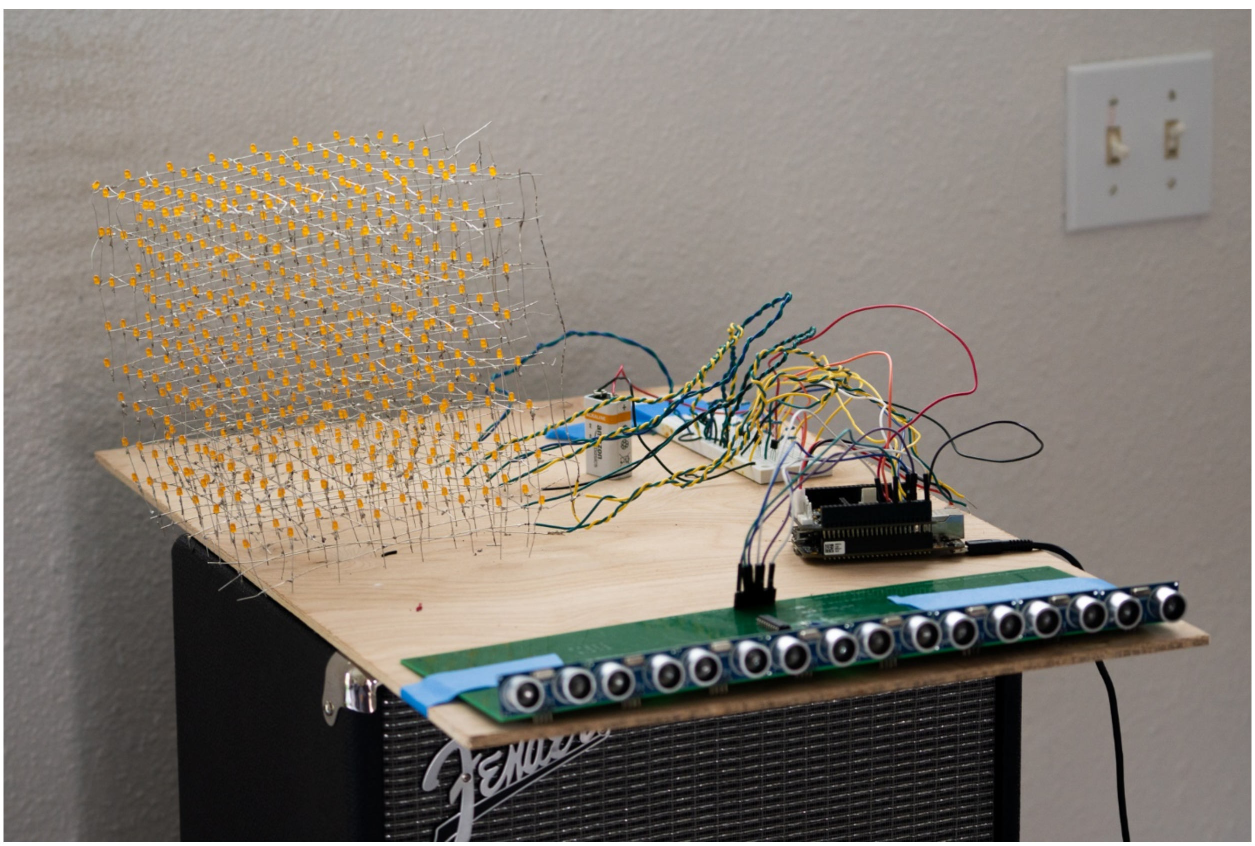


Figure 47: Power Distribution among Project

10 LED Cube:

An integral part of this project is the LED Cube. When implementing the LED cube we must look at a plethora of things, building the LED cube, what LEDs to use, what resistors to use, how to integrate it with the software of Bela, how to power the cube, and much more. The following section we will begin to outline how we would go about building it and the theory behind LED Cubes. Below is a picture of what our cube should look like after it is done. This version however is an 8x8x8 so our Cube will be scaled upwards.

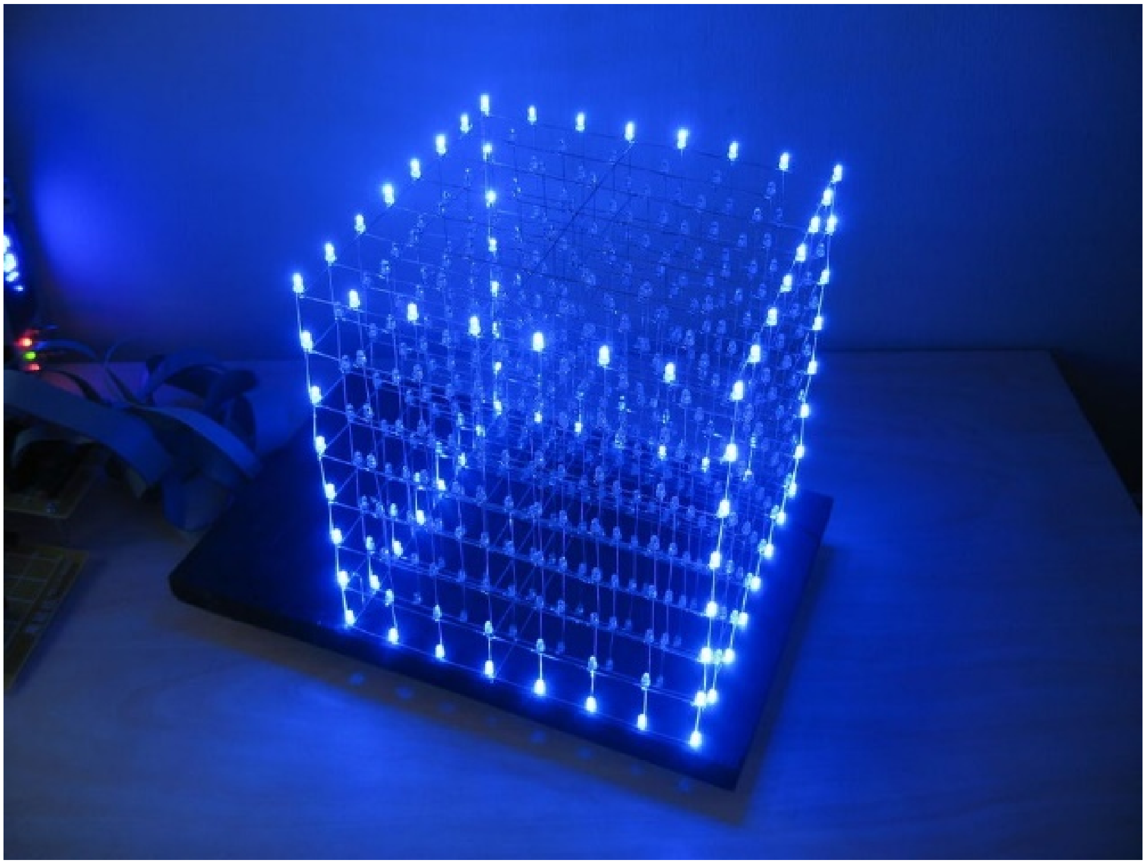


Figure 48: LED Cube (picture printed with permission of Christian Moen)

10.1 Different Considerations for 3D:

Prior to discussing the LED Cube, we should discuss other possibilities for the 3D portion of our project. During our research, we have found other projects utilizing 3D display, those projects have done everything from holograms, to optical illusions using projectors and everything in between. We have decided to not follow these routes for the main reason being cost and personal preference.

Furthermore, if we decided to use holograms we could not have used ultrasonic sensors since by default, using holograms would have required us to have used lasers to scan the images. The reason for this is because holograms work by taking an image and by using light it will reflect that image in 3D across multiple different vantage points. These images are reflected across different mirrors and then displayed, thus causing the image to appear 3D.

Below is an image simply explaining how holograms work:

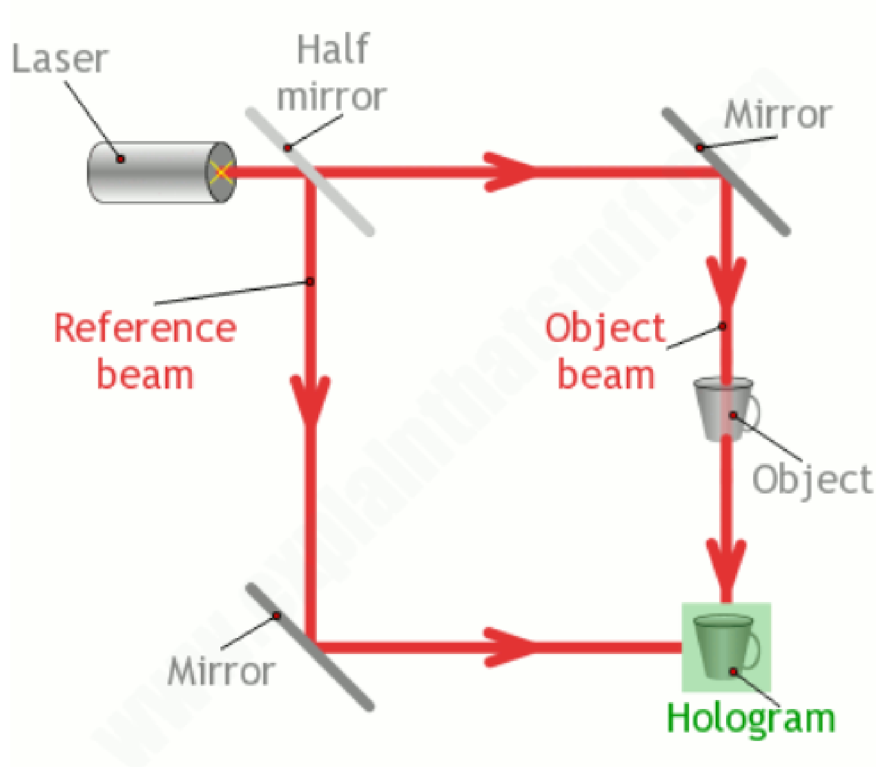


Figure 49

Using holograms would've increased our costs tenfold and would've been a lot more confusing to implement since our group does not have a Photonics Engineer on our team.

When it comes to holograms, many of the technology involved would be considerably above our budget and would be hard to connect via software. Furthermore, we preferably wanted to work with LEDs because we were all interested in utilizing LEDs in our project.

Pictured below is an image of a 3D displaying set ups using holograms:

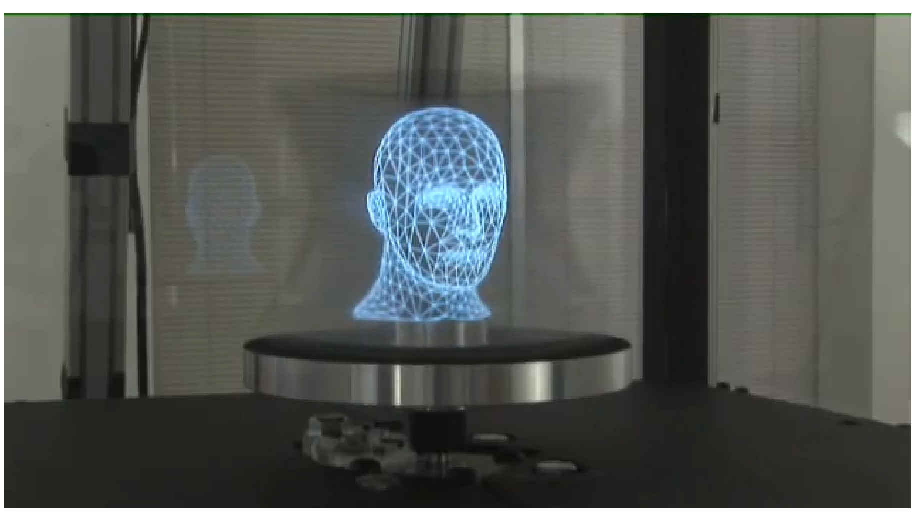


Figure 50: Hologram Example

10.2 Background:

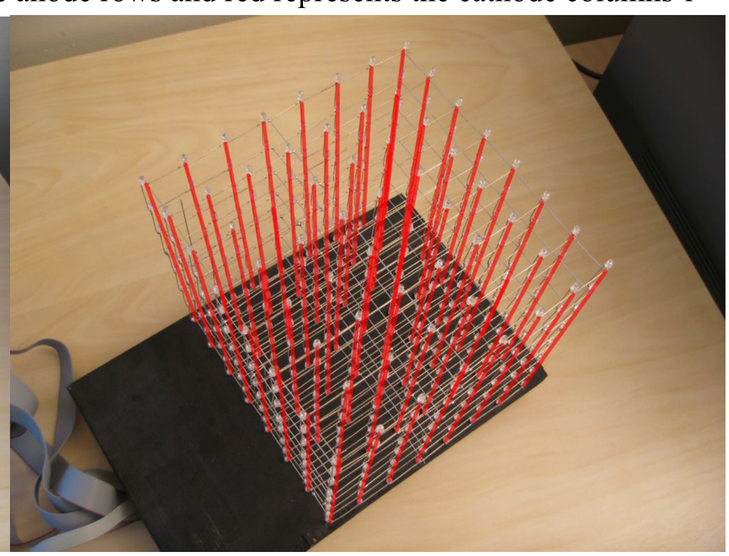
LED Cubes are essentially LED displays but with a third dimension. Because of this, you cannot stack pixels together like you normally would in an LED display, because you must see through the cube which makes it appear 3D. Therefore, you need spacing between the voxels (which are pixels in 3D). Since our LED Cube is going to be 16x16x16, we will need 4096 LEDs to complete this task. Because of the magnitude of building a 16x16x16 array, we will be manipulating an optical phenomenon called the persistence of vision effect (POV effect). This effect essentially states that an image will stay on your retina for some time after it has ceased but only if the image is flickering fast. A great example of this is a spinning torch. When you spin a torch, you'll be able to see a full circle but in reality that is not what's going on. The same applies to our LED cube. When we flash each layer of the cube really fast the image will display as if it is in 3D.

10.3 Persistence of Vision Effect:

The persistence of vision effect is the main reason why we will be able to display 3D images on the LED cube. The brain can only process 10-12 separate images per second and each image can only be retained for up to a fifteenth of a second. Replacing an image with another one in this timeframe will trick the brain into thinking there is continuity between the images. This is how animation works in TV and movies. We will utilize this technique to create the illusion of 3D by flashing the LED layers in quick succession to trick your brain into assuming a 3D image is being displayed. Techniques such as this one has been implemented in many lighting shows

10.4 Anatomy of the Cube:

In our cube, we will be using both anodes and cathodes. The cathodes will be incorporated into the legs (or columns) of the cube while the anodes will be incorporated into the layers (or rows) of the cube. We will then solder the anode legs in each cathode column together and vice versa, each cathode leg in a row together. After that, the 256 columns will be connected to a controller board with a separate wire and each layer will be connected to the same controller board and to a transistor which will turn on and off each layer. The picture above is how the cathodes and anodes will be set up. Blue represents the anode rows and red represents the cathode columns'l

In terms of I/O ports, we will need two sets. One that will turn on the anodes connected to the column and the other to turn off all the cathodes in the layer. For this cube (16x16x16) we will need 256 I/O ports for the anodes and 16 I/O ports for the cathodes. The table below has a list of the different cube sizes and I/O ports needed.

The cube will then be connected to a bottom panel with holes drilled into the panel so that the Cube can rest on top. This will allow the Cube to be steady and not move around as much. Furthermore, this panel will allow us to connect the Cube to the microcontroller

	(x^2)	(x)	(x^2+x)
Cube size	**Anodes**	**Cathodes**	**Total**
2	4	2	6
3	9	3	12
4	16	4	20
5	25	5	30
6	36	6	42
7	49	7	56
8	64	8	72
9	81	9	90
10	100	10	110
11	121	11	132
12	144	12	156
13	169	13	182
14	196	14	210
15	225	15	240
16	256	16	272

Figure 52: Cube Size LED Count

Due to the large amount of current going through the cube, we must ensure that each transistor in every layer can handle it. Therefore, we must take into consideration what color LEDs we use since each one has a different current/voltage needed to operate it. The table below has a list of different cube sizes and their requirements.

Cube size	Leds per layer	Total mA at X mA per LED	
		10mA	20mA
2	4	40	80
3	9	90	180
4	16	160	320
5	25	250	500
6	36	360	720
7	49	490	980
8	64	640	1,280
9	81	810	1,620
10	100	1,000	2,000
11	121	1,210	2,420
12	144	1,440	2,880
13	169	1,690	3,380
14	196	1,960	3,920
15	225	2,250	4,500
16	256	2,560	5,120

Figure 53: Cube Size Power

As a result of the number of LEDs we are going to use, it would be impractical to control each LED with a single wire to each individual LED. Therefore, we will use a technique known as multiplexing. This will allow us to individually control which LEDs are on and off. The part we will use to accomplish this multiplexing task is the TI 74ACT16374. The pin configuration is listed below along with the functions for each corresponding pin.

D1-D8 represent the inputs, while Q1-Q8 represent the outputs. OE represents the output enabled pin. We will use this flip-flop so that we will not have to have a single wire connecting to each single LED. Flip-flops are designed as a basic form of memory, meaning that for any given input the output will remain the same until the OE pin is changed from either low or high.

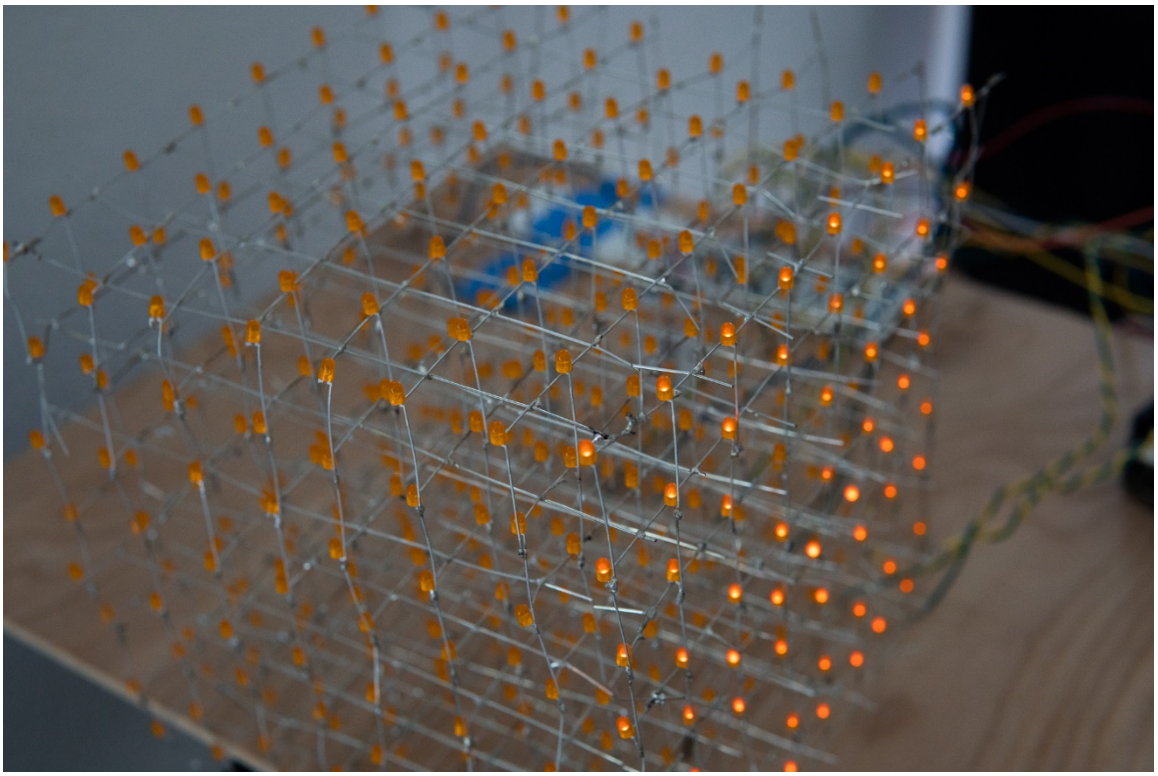


Figure 54: LED Cube

The image above shows how we multiplexed each layer of the cube so that each layer is connected to a certain distance, once the ultrasonic sensors pick up an object at a certain distance it lights up a certain layer based off of the distance. In the image above, the first layer (closest to the sensors) is lit up indicating that the object is extremely close to the ultrasonic sensors.

11 Building the Cube:

Building the cube will be a momentous task considering the sheer number of LEDs there will be. In total there will be 4096 LEDs we will have to work with. Furthermore, there will be over 3000 solder points. Because of this, we will try and make an assembly line between the three group members so that we could test the LEDs, solder each layer, and mount everything. This, along with the sensor array will contain the bulk of the work we will do for the project. The following section will be about the general framework on how we will build the LED cube. Below is a 2D image of a small-scale version of how we will set up the LEDs.

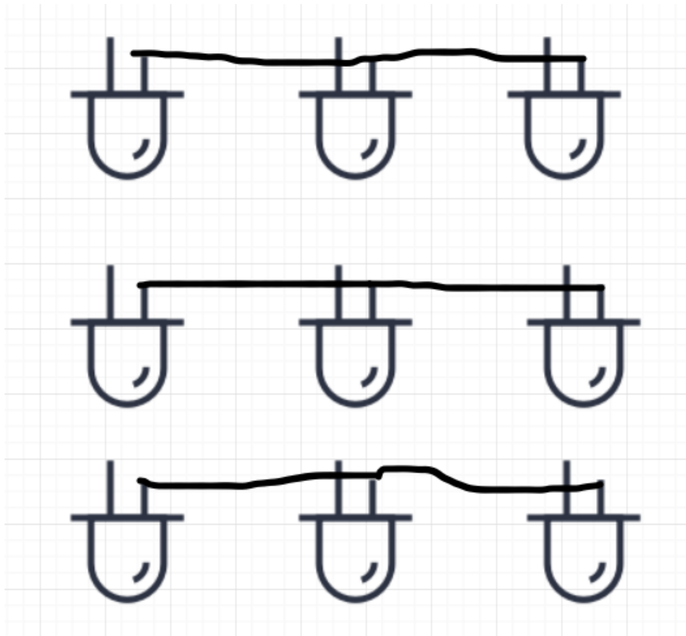


Figure 56: 2D LED Set Up

In the image above, it shows a layer of the cube. Each cathode is soldered together while the anodes are sticking up. Once each layer is soldered together we will take the anodes of each layer and solder them to the layer above and below. By doing this, we will create 16 rows with 16 columns with 16 layers. Thus, creating a 16x16x16 LED Cube. Below are images of our construction process.

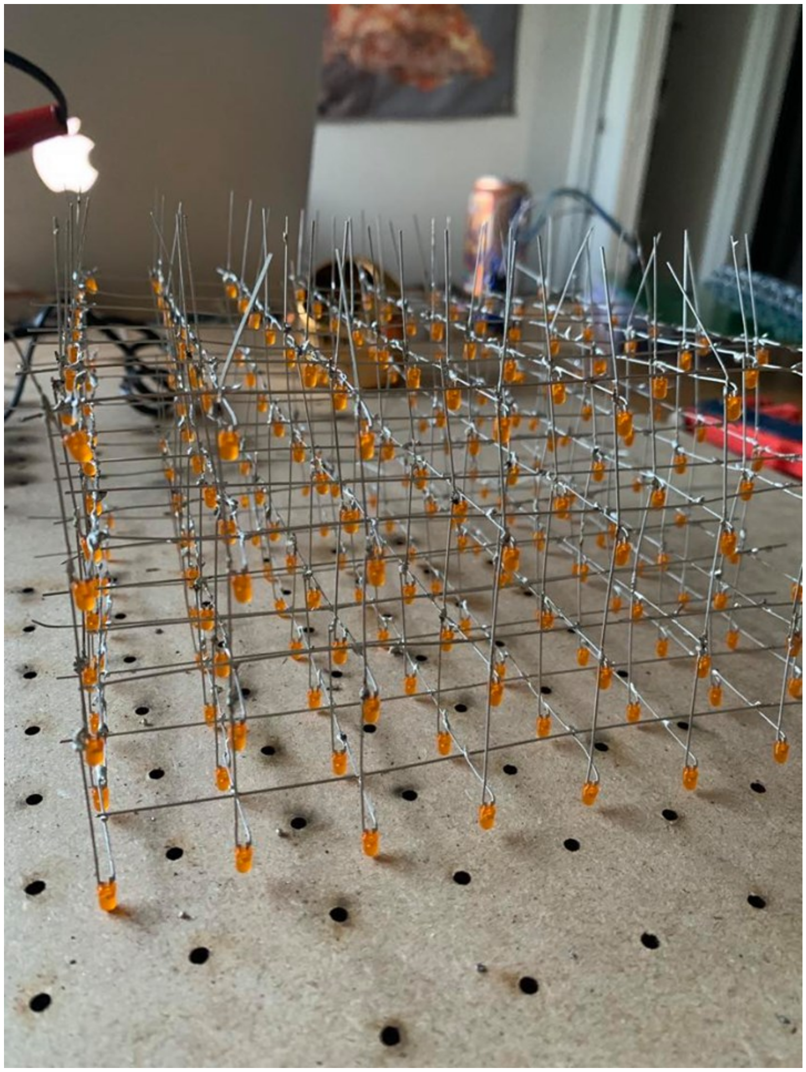


Figure 57: Process of Stacking Layers on the Cube

11.1 Testing LEDs:

Since we are working with thousands of LEDs, we need a way to properly test each LED to see if they are all working. It would be incredibly disappointing to build the Cube and then realize that LEDs were broken or not working properly. Therefore, to quickly test the LEDs we will have each layer set up, and before we solder each layer we will test each LED to ensure that they work, and then test each LED after soldering to ensure that we did not damage the LEDs when soldering. This testing process will take a good amount of time due to the sheer number of LEDs that need to be tested. Utilizing the Senior Design lab offered to us at UCF, we should be able to accomplish this task within a few days.

If we find an LED that is not working properly, we must replace it. To do that, we would have to de-solder the legs from where it is attached to, then carefully pull it out and replace it with a new LED that is properly working. To avoid this, we should try and make sure that each LED is 100% working before we go through the trouble of replacing it.

11.2 Resistors:

We will use resistors that fit the respective values of the LEDs and the 74ACT16374. We haven't calculated what resistor values we will use because we are waiting to order LEDs. However, we will look at values that do not exceed the maximum value of the transistor. Resistors are important for the simple reason that a resistance to low will cause the LED Cube to be blown, and a resistance to high will cause the LED Cube to not light up. Therefore, for the major components of the LED Cube to function properly, we must choose a resistance value for the resistors such that it is the perfect medium for all the components.

11.3 Template and Soldering:

In order to build the LED, we must create a "template" to ensure that each LED is spaced evenly. In order to do this, we will take a block of wood, take a ruler, measure out equal distance, then drill holes in the areas we marked. This will create one layer of the cube. Each layer of the cube will be 16x16 all held together by the legs of the LEDs. The cathode leg of the LED will be connected to the cathode leg of the LED next to it while the anode leg will be straightened 90 degrees upward so that it can connect to the next layer upwards.

Once all the legs are in place, we will begin to solder all the cathode legs together. This will become one layer of the cube, we then must solder the other layers together using this same process, making sure that we do not damage each LED in the soldering process. We will then add braces to each corner to ensure that each layer is secure.

After we begin soldering more layers, we must then begin to connect each layer together by soldering them together. Again, making sure that each LED in each layer is functioning properly and making sure that the LED is properly mounted and not tilting.

An important note we must make when soldering is that it can be very dangerous and can lead to fires, burns, or poison if you breathe in the fumes for to long. As a result of these concerns we will have the necessary safety equipment and gear to ensure we take any and all safety precautions while working with such dangerous equipment.

While soldering we must be able to solder each leg quickly. The reason for this is that prolonged heat exposure will end up ruining the LED. Therefore, it is imperative that we quickly solder and de-solder (when needed) the LEDs. Pictured below is a rough idea for how we want the template to be made:

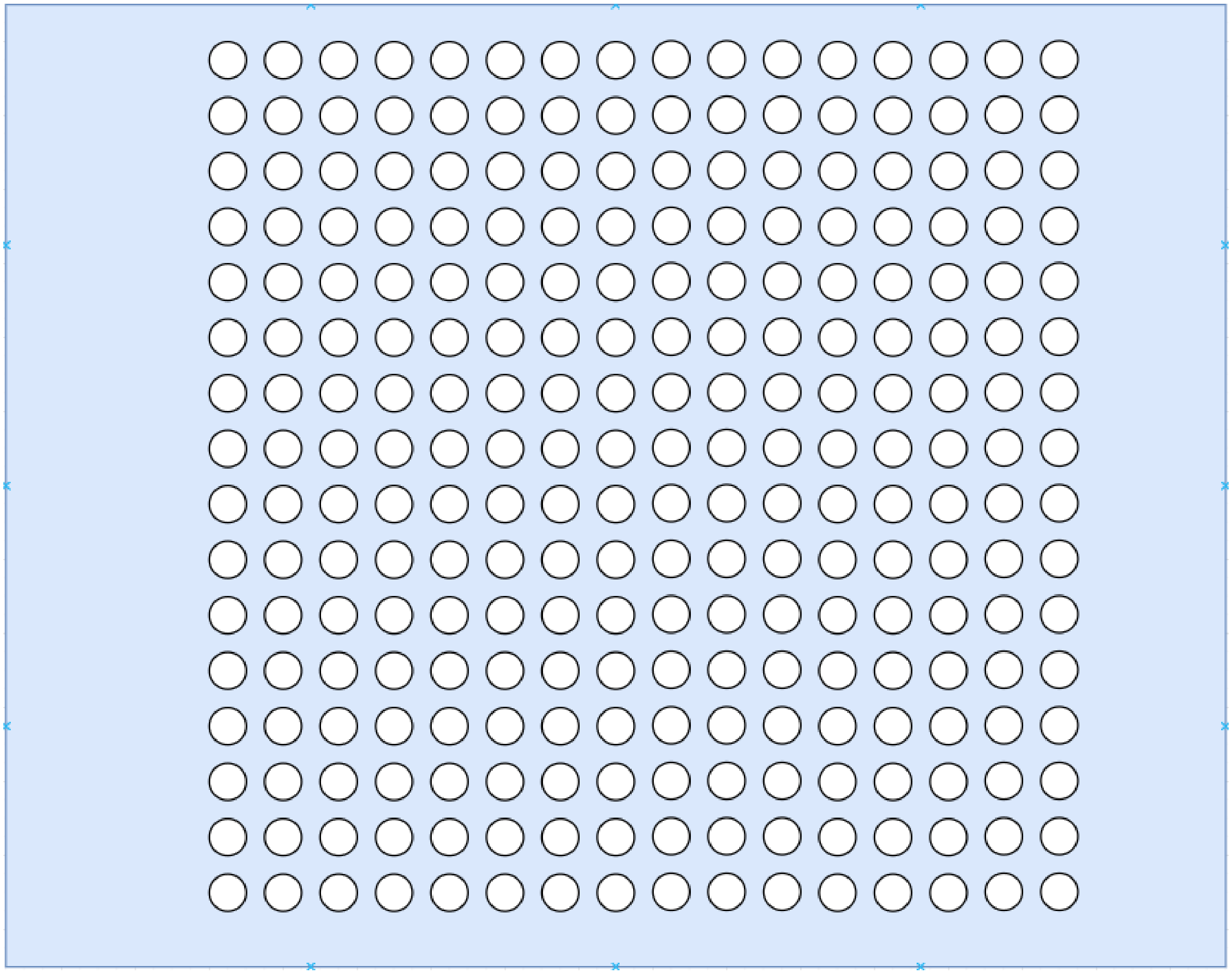


Figure 58: Mount for LED Cube

Each of the white circles represent the spot in which the LED will physically be placed in. Then, each LED will be soldered to the one above it such that it will be at a 90-degree angle. We will then have to make sure the cube is not lop-sided in any way. Furthermore, we will be able to connect ribbon cables to connect to the Bela by drilling under the template (more on that in the next section).

11.4 Connection to the Bela/Mounting:

After the LED cube is properly mounted, we will use the "template" we built to connect the cube to the Bela. We will do this by feeding the bottom layer to ribbon cables connected under the template, which will connect to each wire individually. The ribbon cable will then connect to the Bela via I2C, USB or directly via digital input pins. Below is a representation of how the ribbon cables will be connected:

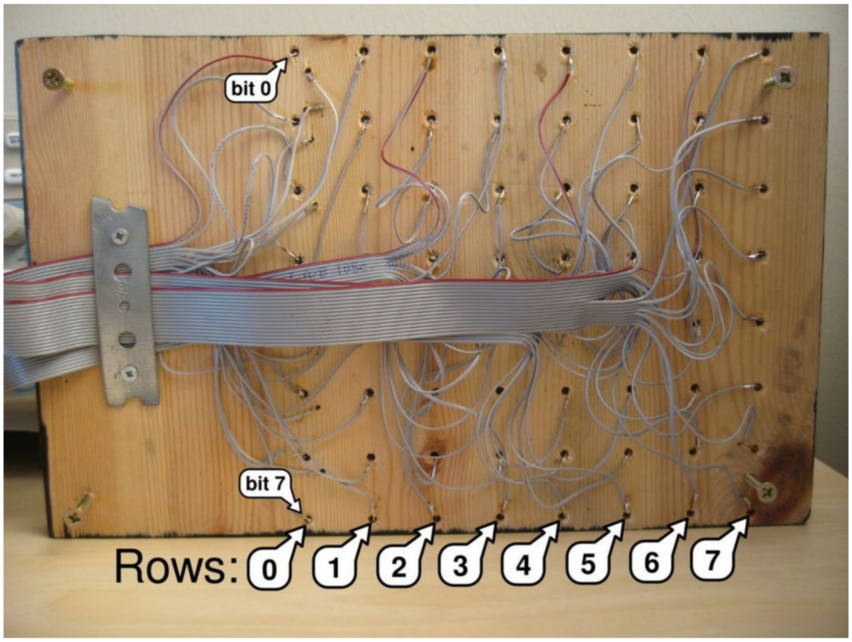


Figure 59: Linking Ribbon Cables to Mount (picture printed with permission of Christian Moen)

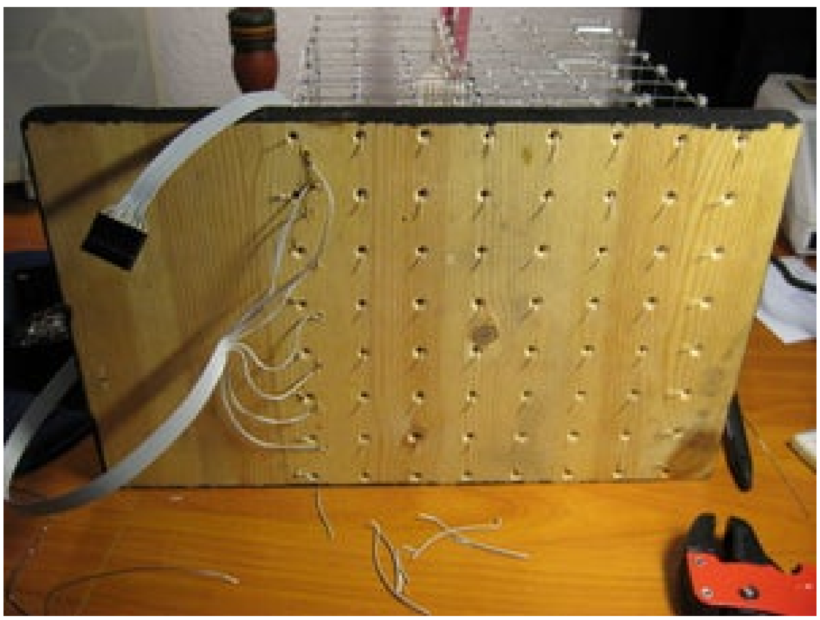


Figure 60: Linking Columns to Mount (picture printed with permission of Christian Moen)

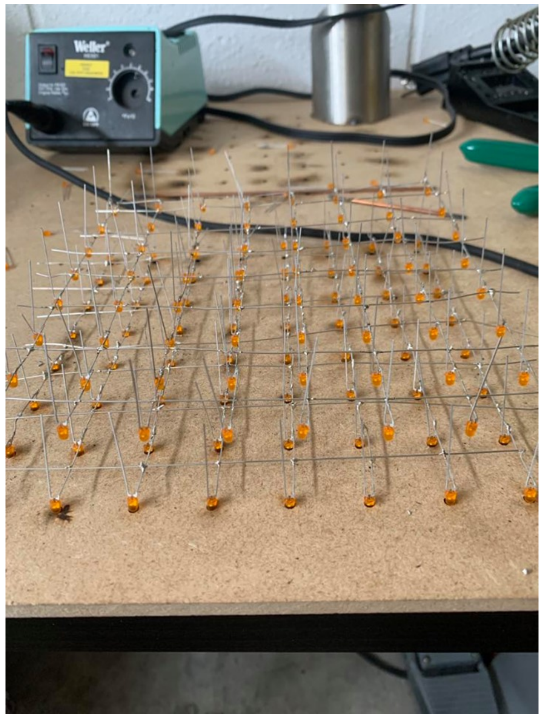


Figure 61: LED Cube in Template

We will then write the software for 3D processing using C++ in the Bela. This software will directly translate 3D images to the LED cube. The software to implement the 3D mapping on the LED cube will be covered in another section. The general format of how the connections will be mounted between all three components is shown below:

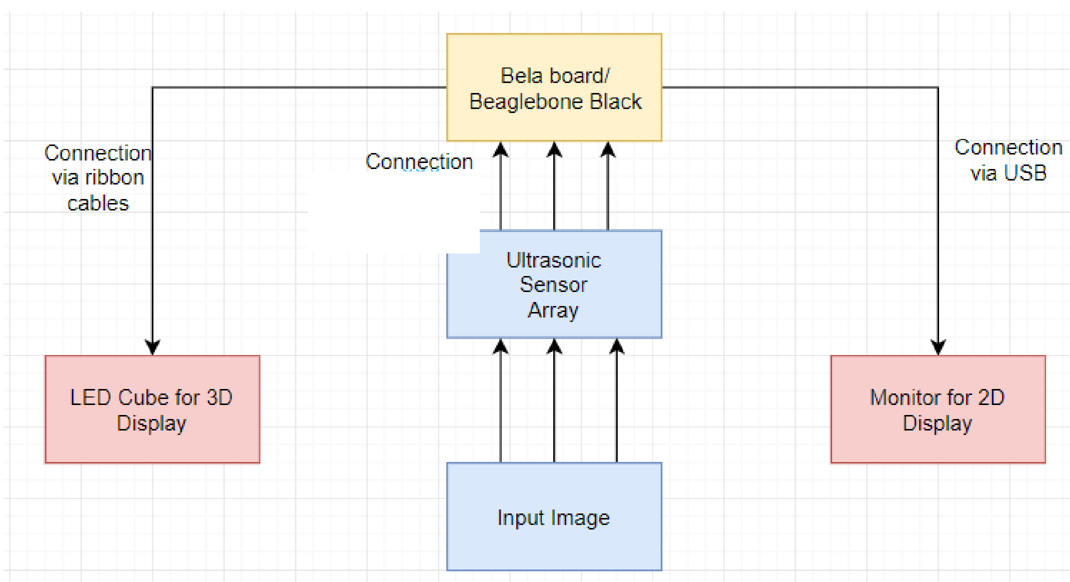


Figure 62: General Layout of our Project

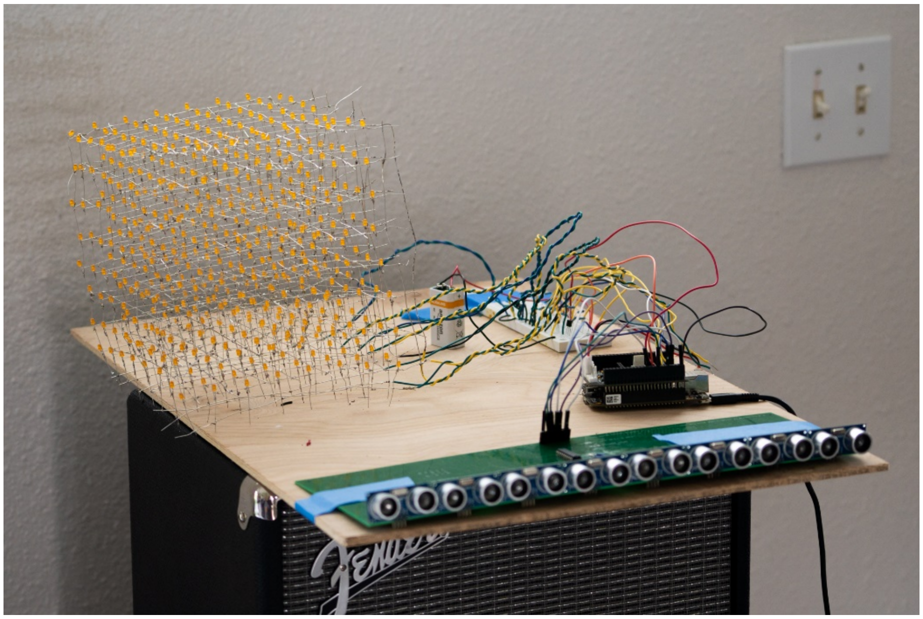


Figure 63: Layout of our Project

Using this general format for our project design is also beneficiary in the sense that we will be able to complete the project modularly. We can split off and do individual components of the project and then regroup to connect pieces and troubleshoot what doesn't work. Because of this, the connections we will have to make using ribbon cables or other connecting devices will be easily achievable due to how modular our project is.

After making all the connections, we will take caution on how we power the board. The reason for this is because if everything is connected, and we use to much power, we could blow our Beaglebone Black, our Bela, and the LED Cube. This could ruin our project and force us to start from the beginning all over again.

After all the connections are made, and the code is uploaded, we will need to debug the entire project. This will be done by testing each component from each section and test everything while separated. Afterwards we will connect each section to another and test the compatibility, surely issues will arise and when they do, we will isolate the issues to see what is going on.

After all the testing and debugging we will then see if we could improve the quality of our project to make it more efficient. This will only be able to happen if we follow the timeline, we have set for ourselves. We will stay ahead of our timeline so that we will make time for improving the efficiency of our project.

12. LEDs:

Our project consists of 512 LEDs. This section consists of the history of LEDs, the process of constructing the cube and testing the cube.

12.1 History of LEDs:

The motivation behind picking this design is due to all of the technical aspects of the design. The design is a cross between various skills needed to create. It is a design that allows for us to utilize all of the things we have learned over the years to become engineer. From sensor array, to LED cube, to even the code that we will have to create to make sure the design works properly. With visual display systems being documented since the 1900s there are many existing projects similar to ours. The first version of the LED cube in my eyes would be the arc lamp. British inventors worked with electric light with this idea. Created in 1835, the arc lamp showcased the first electric light. The lamp tinkered with a bulb that produced light when heated by an electric current. Also, with the bulb atmosphere creating an environment where air would vacuum out or filled with inert gas to prevent the bulb from oxidizing causing it to burn out. The early bulb had very short lifespan, were extremely expensive and also used a lot of energy. The next project would be one which is still used till this very day and is used every day by people all over the world. The lightbulb created in 1879 and invented in 1880 by Thomas Edison and his researchers.

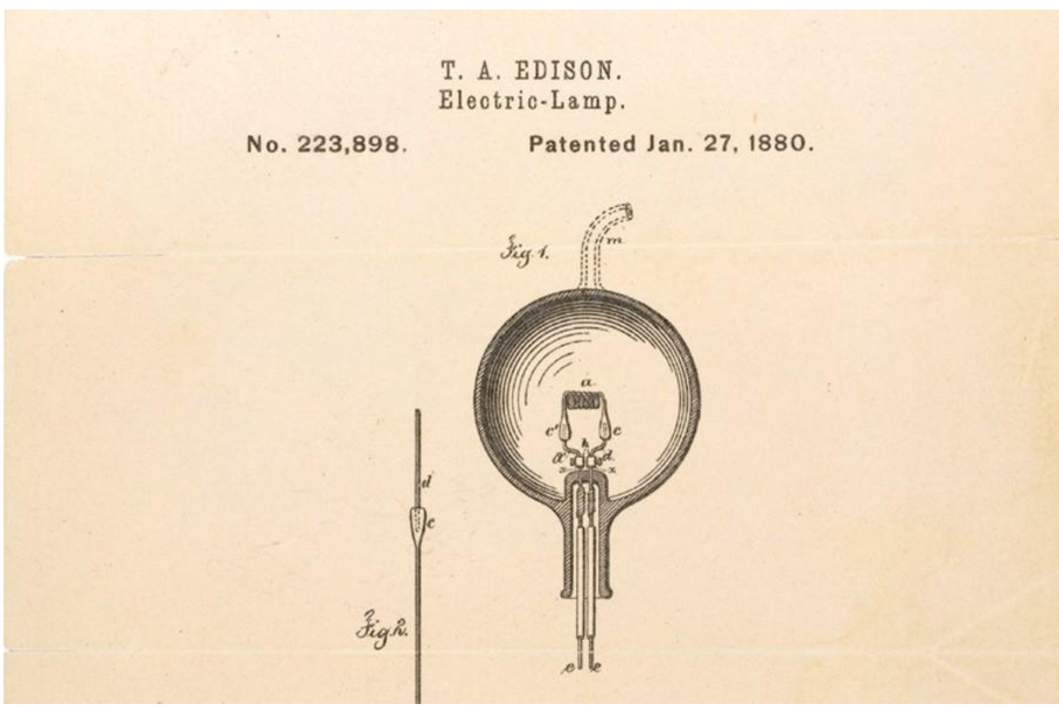


Figure 64: Thomas Edison light bulb invention

The focus of Thomas Edison and his research team was to improve the filament (apart of the bulb). They started by testing by first testing carbon, then for platinum, and lastly back to carbon. In October 1879 is was documented that Edison and his researchers produced a light bulb with a carbonized bulb with noncoated cotton thread which could last up to 14.5 hour. This was a huge improvement from the previous project, the arc lamp in lifespan. Thomas and his researchers continued to work on extending the lifespan until they settled on bamboo which gave lamps a life span up to 1,200 hours becoming a standard for the next decade. This was a very impressive feat and what made it even more impressive is that Thomas and his team continued to work on improving the light bulb technology and lighting system. Showcasing in 1882 that electricity could be distributed from a centrally located generator through a series of wire and tube. This improvement led to the creation of the electric meter. The electric meter is able to track the amount of electricity each customer is using.

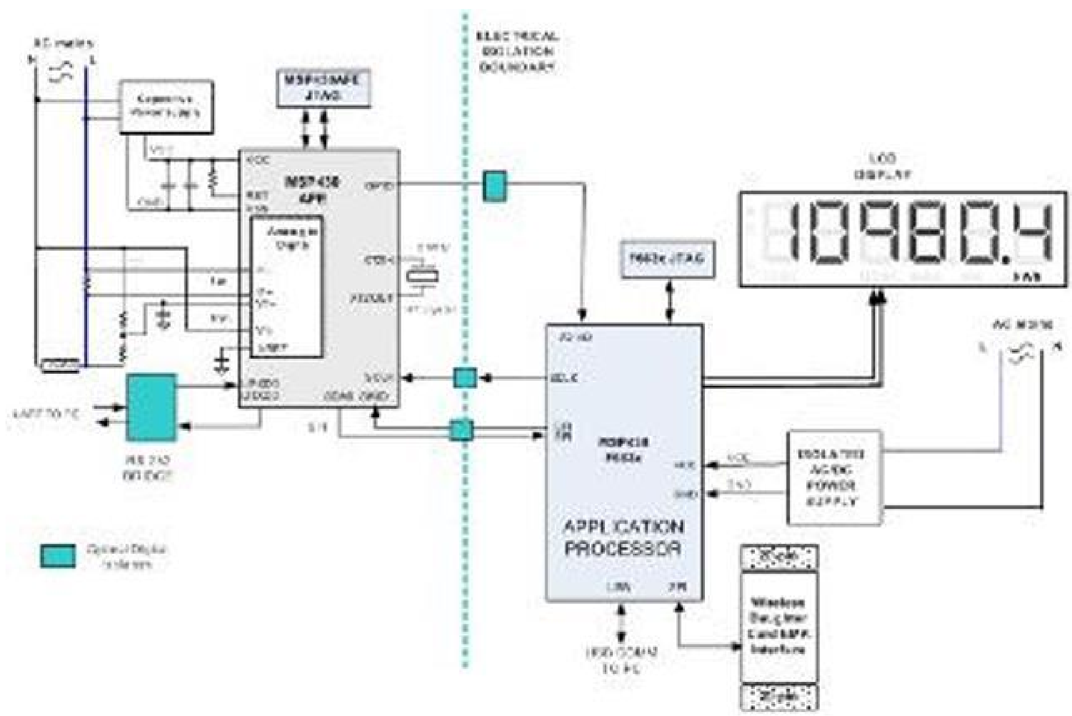


Figure 65: Electric Meter- Single Phase

As time continued more projects and product led to where our project is today. The next project milestone and product in the focus with our design. Due to Thomas and his team of researchers developing the lighting system many other inventors continued making advances and improving the bulb and its efficiency. The next major improvement was the tungsten filament in 1904. The tungsten bulbs not only lasted longer but also was brighter in comparison to the previous light bulb. In 1913 it was discovered that inserting inert gas like nitrogen into the bulb would increase efficiency by two times. More breakthrough in lighting occurred in the 19th century. One of those breakthroughs were the Geissler tube a type of lamp that produced light by removing most of the air from a long glass tube and passing an electrical current through the lamp. These lamps became popular in the 20th century becoming the basis for neon lights, and streetlights which are low pressured sodium lamp. In the early 1900s by passing electrical current through mercury and using a device connected to the bulb regulating the current flow through a tube Peter Hewitt created a blue-green light. By the 1920s many inventor and researcher were using ultraviolet rays and conducting experiments using neon tubes. This wave increased interest in lamps and created programs for lamps across the United States. In 1976 the first spiral shaped fluorescent light bulb was created. These light bulbs would hit the market in the mid 1980 for about 30 dollars. Many problems insued over this invention due to the bulky design making them not fit well into many fixtures including displaying a low output of light with performances varying in many different situations. Since many improvements have been made to the CFL bulb.

The next product that is in focus with our design is the LED standing for Light-emitting diode. LED is a solid-state lighting which uses semiconductor to convert electricity into light. Being small in design and producing light in a certain direction. These lights became very popular being used in Television, Entertainment clubs, and many other appliances. Being the most efficient lights on the market. With light bulb efficiency being measured by light emitted, divided by power drawn LED have the highest on the market. The last existing product that's impact on the lighting industry would be an invention by Nick Holonyak Jr, in 1962 who invented the first LED visible spectrum in red diode. With the invention of green and yellow diode coming after, these inventions ultimately led to us being able to do the design we chose for our visual display system. Without these inventions pathing the way we would not have many luxuries we have today.

For our LEDs, we want LEDs that can be equally visible from each side. In this regard, we will use diffused LEDs since they are equally bright from all sides. Furthermore, we need LEDs that can be easily mounted. So, we will use circular LEDs since they are easier to mount vs square LEDs. Since we are using in essence, an optical illusion to create the appearance of a 3D image, by flashing each layer on and off quickly we will need LEDs that are bright enough to produce a good amount of light in the short time that it is on. Below is the basic anatomy of an LED:

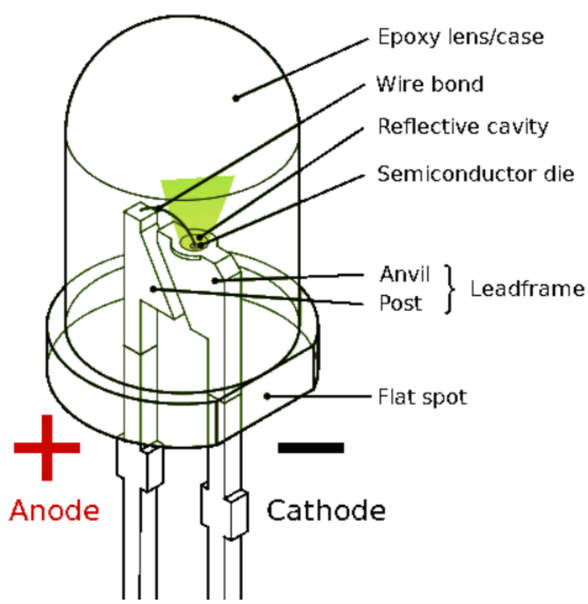


Figure 66: LED Anatomy

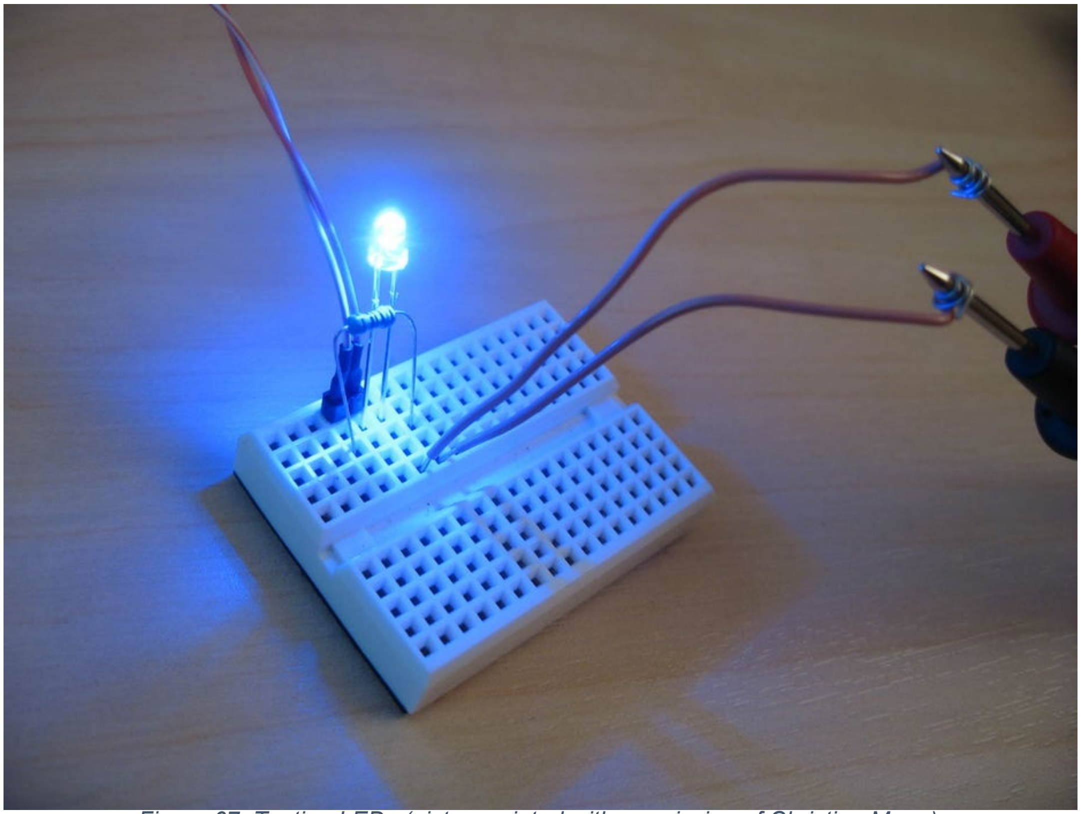


Figure 67: Testing LEDs (picture printed with permission of Christian Moen)

Otherwise, the image will look sloppy. Lastly, the legs of the LEDs must be all the same length since we are going to use the legs to help mount the LEDs. We will also test every LED to ensure that they function properly and are not damaged or not working properly. We will test each LED by connecting it to a breadboard and a resistor and ensuring each LED is properly lighting up. Listed below are some of the LEDs we found on Mouser.

Price (for 5000 LEDs)	**Size**	**Manufacturer**	**Part Name**
255$	5mm	Lite-On	LTL2R3KYD-EM
245$	3mm	Inolux	INL-3AR30
280$	3mm	Lite-On	LTL-4296N

Figure 68: LED Prices Tables

We are still shopping around for LEDs. It is our hope to find a decent vendor soon, however we do have about 500 LEDs on hand to do some small-scale testing of the LED Cube. We have tested each LED to ensure that they are properly functional and that they are fit to be in a small scale LED Cube.

Figure 69: Testing of LED using Multimeter

Furthermore, as shown in the image above it is possible to use a multimeter to test if the LEDs work. Simply set the multimeter to the diode position and then set the positive to the positive leg and negative to the negative leg.

13. Software Integration:

The software we will write is an integral part of our project. Without the software, nothing will work. The basic idea is that the software should be able to process the image that the ultrasonic sensors are reading, translate it into 2D, and then be able to translate it into 3D. Much of the software integration is already supported by Bela and Beaglebone. Therefore, the main thing we will have to do is be able to accurately translate the input to output. The GUI display support for the Bela will also come in handy when it comes to tweaking our project and making the interface look good for presentation.

While the pipeline from sensor array to LED cube seems to be, on paper, a simple task of plug and play, this is far from the case. Considerable effort must be put into designing a software package capable of not only processing the incoming data stream, but both rendering for 2d *and* repackaging to be projected onto the LED cube. Thankfully we have designed the hardware side of our data processing sections (multiplexer units) to streamline the task of handling the relatively large amount of information. One that, hopefully, should be able to handle the operations in a manner that nearly emulates real-time video output.

As previously stated, the system employs a method of demuxing to group the sensors into sets small enough to be directly read into the microcomputer. By modulating which groups are powered on via a binary selection line, we are able to evade any dicey situations presented when trying to digitally demodulate a continuous analog stream. The general structure of our software suite is wholly informed by the way we have set up data collection and output as, with any embedded system, designing software for an unknown hardware system is nearly impossible.

From a top down level, the software, written mostly C++, must first handle the selection of groups (via modulation of 2 output pins), the collection of incoming data from said groups, and the tagging and filing of the incoming data into X,Y, and Z tables are a must. It becomes obvious that a set of arrays need to be developed in order to store both the positional data of the reading sensors and their respective depth measurements. In order to do this with some ease (and to better use the optimal parts of C++) structs can be developed to define a kind of multi-dimensional array that is then superimposed onto all the groups.

As each group will be inputting into the same pins, (multiple connections per input pin but with only one "hot" connection line), we will need some way to delineate between what sensor is actually being read to a specific pin at a given time. To remedy this, we can track the incrementation of the group counter.

The groups must be changed at a fast-enough rate to grab a full frame of data in a manner that will feel fluid and generate a smooth image, ideally 32 groups at 24 frames per second yielding a group refresh rate of 768Hz. While we increment what the sel line we can track what group number is currently being accessed and modify the data held in the group structure to pull the respective coordinates of the sensors being read at the instant of reading. As we are dealing with a permanent internal hardware installation, we can simply hardcode each group respective position data and use the SEL lines counter to change what group is being accessed.

Once we have marked which group is being accessed and have an understanding of the physical locations of each sensor within the group. We can log the respective data to a 3-dimensional array holding the respective X and Y data, and then pushing the updated Z-depth data to the core of the array. This methodology develops an easily manipulatable object containing a real time representation of our system in 3D.

For example, if we are choosing between two groups, we first check which group is being accessed, then update the array values that will be logged too. Then we simply add to the SEL line controlling the group, and parse through each group, logging the incoming data at each step. This methodology would require the use of hardcoded X,Y values for every group and would also require the implementation of multiple multi-dimensional arrays, something that could easily get confusing to deal with especially when real time processing is a concern (potentially longer access time for these kinds of complex data paths).

Instead of utilizing a three-dimensional array, we could also log the data to a single 256 slotted array of floating points and number them in a way that allows us to quickly read the data. There are two obvious ways of numbering the sensors in this array. Simply increasing the value from top left to bottom right would suffice but a potentially more useful, or at least more novel, way would be to utilize the dimensional compression qualities of space filling curves.

13.1 Mobile Applications

At the very beginning of our planning stages, when deciding what to do and how to do it, we decided early on that we would not incorporate mobile apps with our project. The reason being is that they provide no logical benefit to our project. As a result of this, we have decided to not move forward with developing mobile apps with our project. However, it is still necessary to provide an outline of mobile applications, their benefits and cons and if we were to implement them, how we would go about doing it.

The mobile app we could've developed would be able to integrate itself with our device. Our original idea was to have it so you could upload images from your phone (or take images with your phone) and then have those images displayed on the monitor in 2D and the LED Cube in 3D. However, this idea would introduce a whole array of complex issues involving security, network, compatibility issues. Therefore, we decided to scrap the idea all together.

On the other hand, there are many benefits to developing mobile apps, for one it allows you to reach a broader audience. Furthermore, phones are extremely popular, therefore it further expands your compatibility with more systems. Allowing yourself to integrate and be a part of a plethora of systems allows the developer to become a staple of technological force.

Mobile applications come with a plethora of security risks such as someone hacking the app and by doing so, having access to all the users of the app's personal data. However, this can be mitigated by implementing a plethora of security measures to ensure information stored within the app is protected.

When considering what platform to develop mobile apps, it's important to understand your demographic and understanding the audience you're trying to reach. Also understanding which platform has more of a reach. iOS has more of a reach than Android. Therefore, it would make sense for us to develop our project on the iOS platform. Below is an image showing the percentages and sales for both platforms.

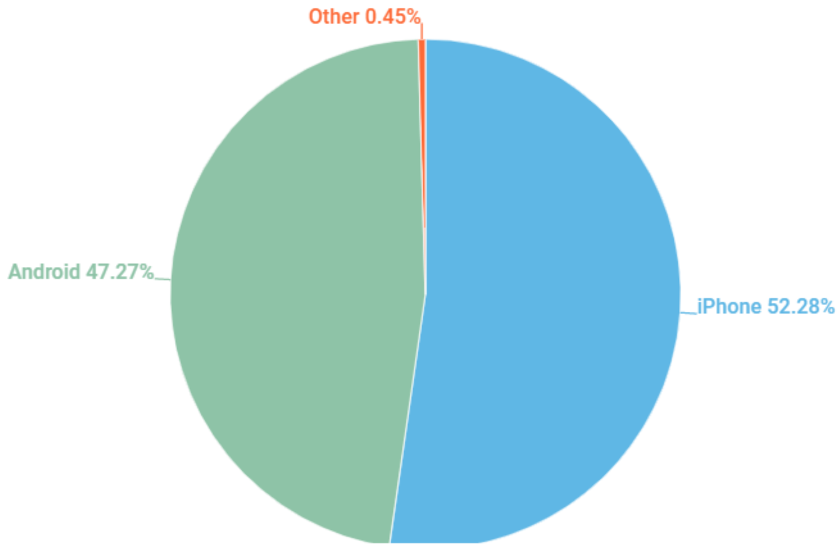


Figure 70: iOS vs Android North America Shares

This graph shows that if we were to develop a mobile app, iOS would be the way to go since we would be able to reach more people. However, doing both Android and iOS would be beneficial due to the fact that globally, Android is used more.

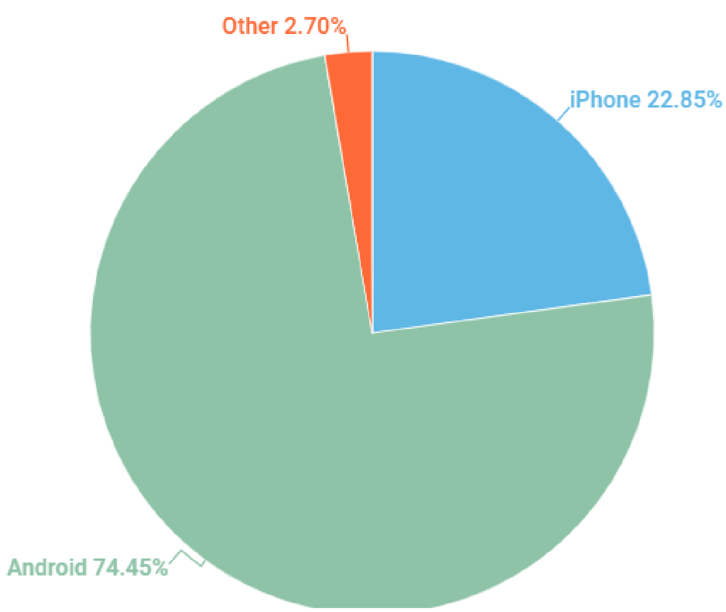


Figure 71: iOS vs Android in Europe

13.2 Languages

As mentioned in previous sections, we are using C++ to implement our project. However, there are a plethora of languages we could implement for our project. Languages such as C, Java, JavaScript, Python, etc. All carry their pros and cons. Below is a table comparing the possible compatible languages with the Beaglebone

Language	Paradigm(s)	Standardized	Type Safety	Expression of Types	Type Checking	Parameter Passing	Intended Use	Design Goals
C	Imperative	Yes, ANSI C89, ISO C90/C99	Unsafe	Explicit	Static	By value reference through pointers	System, Embedded	Low level access, Minimal constraint
C++	Imperative. OOP, Generic	Yes, ISO C++/C++11	Safe	Explicit/ partially implicit	Static /Dynamic	By value, reference through pointers	Application, System	Abstraction, Efficiency, Compatibility
Java	Imperative. OOP, Generic, Reflective	Yes, Java SE Specifications	Safe	Explicit	Static	By value	Application	Write once run anywhere
Javascript	Imperative. OOP, Functional, Reflective	Yes, ECMAScript Standard	X	Implicit	Dynamic	By value	Client-side Web Scripting	X
Python	Imperative, OOP, functional, aspect-oriented, reflective	No	Safe	Implicit	Dynamic	By value (Call by object reference)	Application, education, scripting	Simplicity, Readability, Expressiveness, Modularity

Figure 72: Table of Languages

Furthermore, having C++ as our main language means that if we were to ever build a mobile app, it would be not to difficult to port. The reason for this is because one of the languages that iOS's development kit is Objective-C. However, for Android development the language they use is Java. This will make it somewhat difficult (but not impossible) to port. It is our hopes, that if we were to ever get funding, we would begin to increase the development of our project in terms of scale and develop a mobile app.

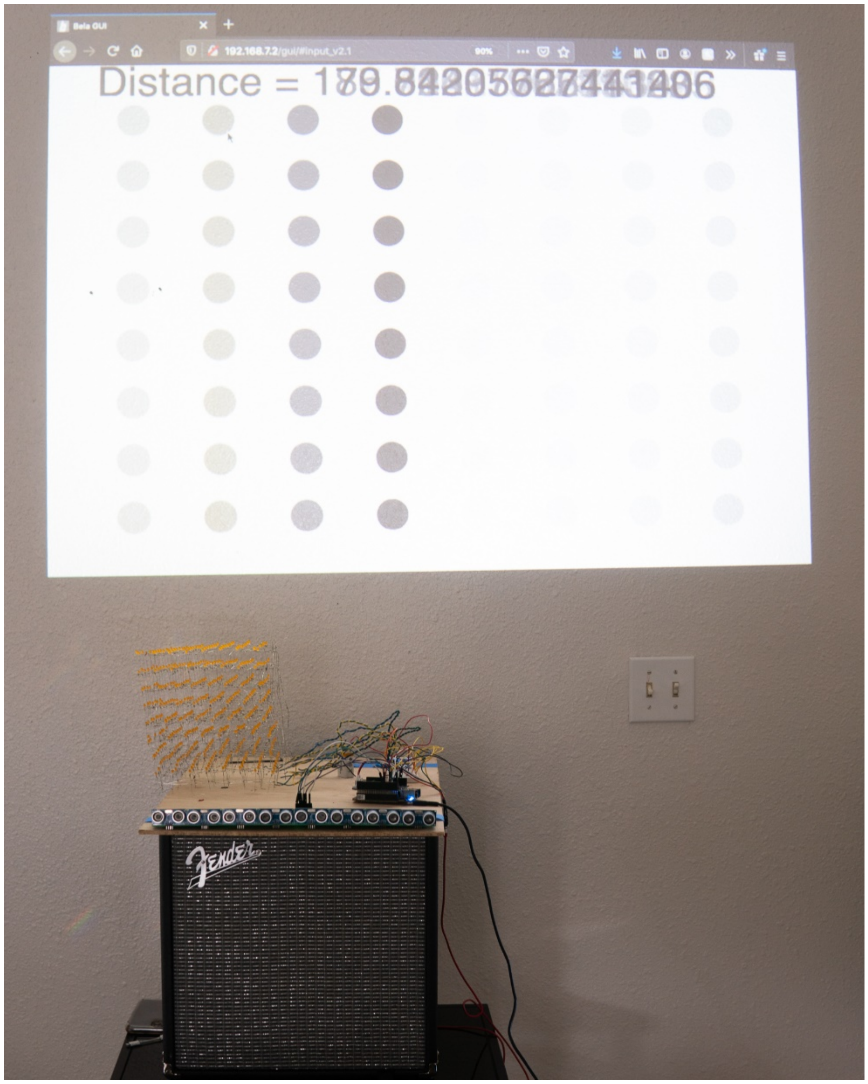


Figure 73: Software Component Complete

The image above shows our fully completed software GUI. It takes the input that the ultrasonic sensors are feeding it and displays it in a GUI.

14. Hilbert Curve and Dimensional Compression:

If we were to use a simple incrementation from top left to bottom right we would run into some issues of locality in regards to the way our array is grouped. For example, if we are to call the top left corner of a square grid, “0”, and the bottom right corner, “16”, we would find that, upon incrementing from left to right downward, that the distance between two points on the 2D grid would not be similar to the difference of their numerical values (or more pertinently the distance on the one dimensional curve). While not an inherently problematic situation to be in, this does diminish the readability of our data. And, as such, when dealing with large quantities of information whose intended use is to be visualized, being able to quickly comprehend data sets is of utmost importance.

Understanding the usefulness of an easily readable and manipulatable one dimensional data set a realization occurs; if we can utilize a method of distributing a one dimensional array across a 2d space we could drastically decrease the overall code complexity and increase the legibility of our system at next to no cost. But how can we implement this?

Enter the Hilbert Curve. This is a type of space filling curve with a unique property, its conservation of locality between 1D and 2D spaces. This curve is a special variant of the space filling Peano curve discovered by David Hilbert in 1891. By constructed in steps that when constructed an amount of times will pack to touch every point of a square form. Hilbert Curves are a special case in which locality conservation is maximized. In short, the Hilbert Curve is an optimal curve to utilize when one wants to describe two dimensional sets of data in a single dimension while maintaining the positional locality of each data point. This allows it to be an incredibly useful tool in dealing with image processing as one can simply describe a whole 2 dimensional set (and in theory any nth dimensional set) of data as a one dimensional form whose general inter set positional data is maintained throughout the curve.

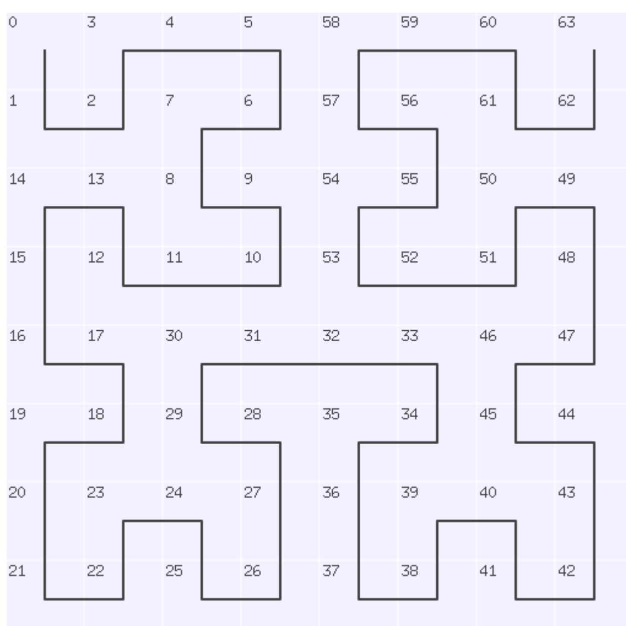


Figure 74: The Hilbert Curve mapped to a discrete grid, note the continuity of localization.

The direct implications of employing a Hilbert Curve in our system are profound. In being ingenious with our numbering system and forming it as a discrete variant of the Hilbert Curve we can fully evade the use of multi-dimensional arrays to describe the incoming data in lieu of a cleverly enumerated one dimensional array. Not only does this reduce the overall complexity of managing our software but this allows us to simple shift further down the array and modulate the size of our groups. In effect, if implementing a Hilbert Curve based array for data logging, we can log to the first 8 slots of our 1D array, increment the group counter, and move up the array. From a mathematical standpoint this yields a simple equation for determining what group is being logged, and the relative inter-group sensor positions.

$$G[A,B]=[8n,8(n+1)]$$

Where *n* is the current group number, *A* is the starting position of the logging set and *B* is the ending position of the current logging set. Increasing *n* per each group will move the logging set along the one-dimensional array and log them in each group. Considering that we will reference this data path in terms of the Hilbert Curve, we have generated a decently elegant solution to the issue of locality and data logging.

An additionally useful function of the Hilbert Curves locality conservation property is that it radically increases the overall scalability of the project. As spatial locality is increasingly conserved throughout each order of the Hilbert Curve, we can effectively scale up the system to be of an infinitely dense resolution while only having to make minimum changes to the overall mappings of our codebase. This is due to the limit placed on locality shift by the Infinite Hilbert Curve. As such our system could scale to be larger and larger and the relative XY to T mapping (2D-1D) of any given point would begin to change less and less. As our points are inherently discrete this change in locality would be further reduced and thus mitigated. Obviously, this benefit is only maximally realizable when using an incredibly dense sensor array, but one does begin to see benefits, even if small, in the smaller orders of the curve.

14.1. Treatment for 2-D Representations and Halftones:

Upon properly processing the incoming data stream into a taggable set of data (whether via multi-dimensional arrays or a Hilbert Curve based dimensional compression scheme) we are now able to visually represent our data. The Bela IDE comes equipped with a power GUI feature set that supports JavaScript. By utilizing our newly generated referenceable data set we are now able to assign these data points to be visualized in 2D.

A simple setup for this is to create a 16 x 16 grid of circles whose XY positional data is linked to the data of the sensors. As this is a 2D representation we are only creating 16x16 circles and thus will only be generating depth *perception* instead of an actual representation of a 3D image. Once the grid of circles has been deployed, we can use each circles respective Z data set to modulate how big or how small the specific circle is. The composite of these smaller and larger circles will enable us to create an image that accurately simulates the depth of a real object.

The scaling of an objects perceived size to distance is not a directly linear one, and to boot there are several interpretations of how exactly one should represent the relationship between size and distance. A cursory search yields two strong contenders for a realistic relation in size scaling; *1/x* and *Arccot(x)* to be specific. These are extremely similar graphs, especially in their scaling of images at a distance, but the choice is really left up to what ends up looking the best when properly implemented. (General consensus seems to say that 1/x may be more accurate to real life).

Creating images as a composite of loosely connected dots is a common practice, especially in the screen-printing world. Much like how we are currently limiting our image output to be of only one color and utilizing size discrepancies to create the image, many screen printers have specialized in limited color demonstrations of lighting and depth emulation by using dots patterns. This effect is commonly referred to as "Halftone" printing.

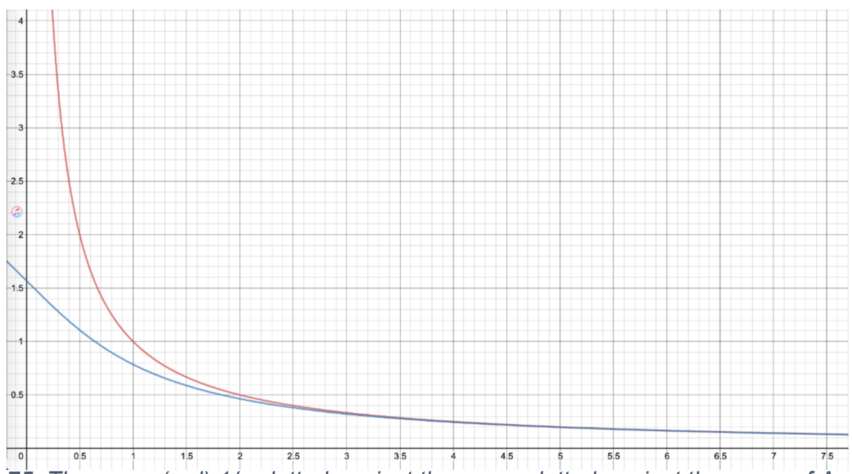


Figure 75: The curve (red) 1/x plotted against the curve plotted against the curve of Arccot(x)

While obviously not considering true depth and focusing more on the effects on depth perception that light and shadows create, Halftone printing shares a strong resemblance to our 2D visualization. By increasing the size of each dot of a halftone print, one is able to easily represent variations of depth and lighting that would be otherwise lost on a simple single tone image. This demonstrates a seemingly apparent theme with almost all facets of image processing, the data represented needs to be multi-dimensional enough to properly represent an incoming image, and the more dimensions of data an image represents the closer to real life said image becomes. As is indicative with our own project and with halftone printing, these dimensions do not need to be color and lighting but can be things such as modulations of pixel/voxel/dot size and changing
distances.

As a continually forming heuristic, that may be obvious but none the less potent, the quality of an image is directly linked to the number of dimensions its data takes the form of. A single line of monochromatic pixels conveys far less information than a full color painting which in turn is far less than a full sculpture. The upper limit of this would be a perfect simulacrum of real life with all its trappings and experience fully described, IE an image with an infinite amount of data dimensions would be indistinguishable from real life and be the highest form of simulacra achievable.

14.2 Packaging Data for Output:

As is quickly becoming a common theme while developing image reconstruction systems, the methodology used to record and tag incoming data is the reverse of that used to reconstruct said data. As we are looking to use a similar pin configuration to actually output to the cube unit we need to simply reverse the order of our initial data collection stage. Upon being input into the Hilbert array we can read from the array one group at a time, writing to the output pins, and then increment the group counter value to move the read window to the next groups worth of pins. This constant shifting and reading of a 1D array should make for a lightweight streaming system. Additionally, if we start the initial reading a group behind the one being written too, we can tie the read speed to the write speed, guaranteeing a smooth nearly real time pipeline.

The group value can be directly output to the cube as well, turning on and off each individual set of LEDs. This is necessary as, like in the input stage, there are multiple connections coming into each pin but each pin should only have one “hot” connection. Unlike the active transducers, however, the LEDs do not have a specific power supply pin and we may need to employ some form of transistor switching or relay system in order to properly control which group is being accessed at a given time.

This hybrid form of parallel/serial communication has obvious benefits, for one we are able to display multiple data points at the same time, speeding up our overall refresh rate and adding the image quality immensely. Additionally, we have an incredibly decreased amount of necessary I/O, effectively achieving a perfect balance between the condensed size of serialization and the massive throughput of parallelization.

The general workflow for our internal software route is described in flowchart form below. While clearly simplified, this serves as a general diagram of how our C++/JS program will be operating. It is more or less a clear shot from input to both outputs with a few loops checking when groups have been recorded/output to push the next group array in. Some abstractions will be required to properly apply the Hilbert Curve to the code base but, as this is a proof of concept and the Hilbert Curve at its 4th order is only 256 points of data we may opt for hardcoding 2d to 1d mappings. While we could opt to dynamically allocate Hilbert Mappings, the work required to make a theoretically infinitely scalable set of these mappings would be a task worthy of its own report and thesis. As such, simply directly mapping these points/conversions between 2D-1D cartesian data should suffice for our use case.

By using the Hilbert Curve, we will be able to accurately map out which ultrasonic sensor goes with which LED. This will increase the efficiency of our entire project and reduce the scale.

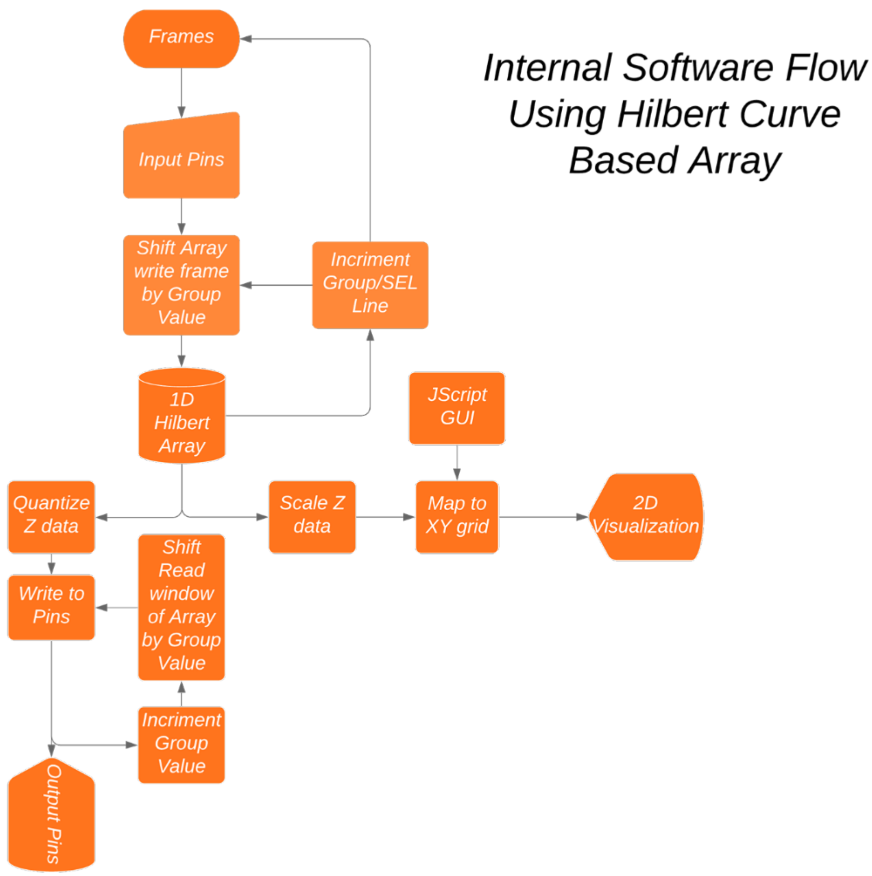


Figure 77: Software Flow Using Hilbert Curve Based Array

15. Project Operation

This project will operate by receiving an input via the ultrasonic sensors, (for example a hand) then displaying that hand in 2D on a monitor, then in 3D on an LED Cube. It is our hopes that this model/design will be scaled up in the future and used for music festivals or big events. This device is purely used as entertainment and the following section will outline troubleshooting tips, safety precautions and general information.

15.1 Safety Precautions

This device is a completely safe device. However, with any project, things can go wrong, therefore safety precautions must be taken when operating the device to ensure that all safety measures are taken.

1. Do not operate the device near any liquids to ensure the device is not damaged and no electrocution occurs.
2. Equip insulator gloves if work on the project must be done on the spot. This will prevent electrical shock.
3. Ensure that no wires are exposed, loose, or exposed to the elements. This will help prevent degradation and electrical shock.
4. If the situation arises where the device must be tampered with, please shut off the power and let it sit for a few minutes to ensure the device is not holding any electrical charge.
5. If any of the LEDs must be replaced, ensure that proper solder safety is followed.
6. Ensure that you are in a closed room so that you can properly see/use the device.

Following these safety precautions are important to ensure that no one gets hurt operating this device.

15.2 Using the Device

In order to use the device, ensure that all the connections are firm and properly made between the Bela, Beaglebone Black, Monitor, LED Cube, Ultrasonic Sensor Array, and any other connections that are made. Secondly, ensure that the software to properly render in 2D and 3D is up and running and working properly. Thirdly, ensure that the Ultrasonic Sensor Array is accurately receiving input. Lastly, give the Ultrasonic Sensor Array input for it to display. After the input is received, the software/Beaglebone Black will process the input and then render it in 2D then render it in 3D on the LED Cube. Before using the device please ensure that all safety precautions are taken

15.3 Troubleshooting Tips

Listed below are a few tips and tricks for common troubleshooting issues you might have. This table will allow the user to fix common issues they might have when using the device. It is our hope that this section provides immense help and reduces the complication of a device that is intricate and complicated.

Problem	**Troubleshooting Tip**
Device not powering on	• Make sure the device is receiving power • Make sure there are no loose connections/wires
Device is not outputting correctly	• Make sure the LED Cube is working properly • Make sure the software is working properly • Make sure the Sensor Array is working properly
Ultrasonic Sensor Array is not picking up the correct input	• Ensure the target is at a reasonable distance from the Array. • Ensure that each sensor is working properly
LED Cube not responding to input properly	• Make sure each LED is working properly • Make sure that there are no loose connections on the LED to the Bela/Beaglebone

16. Conclusion:

With most people usually being familiar with image reconstruction via optical sources. Our optical reconstruction conferred a great way to simulate the human vision as able to capture the full spectrum of color, and no exceeding at depth and structural reconstruction. Since we were willing to exchange color for a high level of depth awareness, we ended up with a sonic approach that became available. The sonic image reconstruction having two major advantages over optical imaging, low light operationality and fine readings of true depth which were also opposed to a simulated "perception" of depth. By sending ultrasonic pulses outward and calculating the round-trip travel time a fine reading of the distance from the sensor was generated. This allowed for sonic imaging to be able to accurately represent depth perception, albeit at the cost of any form of color. With this form of image reconstruction performed constantly in nature, most infamously by bats. The ultrasonic echolocation implored by bats allowed them to obtain an understanding of their surroundings in low light situations that lets them navigate caves while flying, and thus is of high precision and capable of generating a high-fidelity reconstruction of the surroundings. In our experiment we used this same feature of image reconstruction to give us an output image ideal to what we wanted our visual display system to display.

The 3-D image reconstruction was difficult to create by the visual display system. Being displays are rarely capable of legitimately recreating three dimensional images as they are themselves two dimensional. As such Frames requires a voxel based 3-Dimensional display to be able to properly complete the imaging process. A cube of XYZ addressable LEDs must be deployed in order to properly construct the image that was taken in at the inputs. In designing this cube, we are able to properly show the totality of the incoming data stream and properly display depth as opposed to a simply simulating depth *perception*. The Real Time processing and outputting of this incoming data stream is processor intensive and required a large amount of I/O ports, as such we opted to use a powerful and expandable microcomputer to easily handle the imaging process. Specifically, we have gone with the Beagle Bone Black for its strong GPIO selection and plentiful RAM. After applying the Beagle Bone Black, to our design we can produce our visual display system by incorporating the various sensors together to create a 3-Dimensional display able to properly complete imaging processing.

Along our journey of designing this system a common theme cropped up in regards to image reconstruction. Namely, the concept that image capture is the inversion of image reconstruction. In lay terms this seems like a painfully obvious heuristic but much like any great adage, its simplicity begets a deep understanding to fully comprehend. For every action we took to process, file, and record the data from our sensor array; the exact action in reverse must have been taken. This simple yin and yang was at first difficult for us to understand. We kept trying to design our input system and our output system differently, continually making the mistake of thinking that we had to have two vastly specifically and unique systems to handle output and input. After hours of deliberation over various I/O schema for images, it would become clear, at every level of our device, a simple realization. Effectively being the Occam's Razor of image reconstruction we found that more often than not, the best input scheme is also the best output scheme. Once finally reaching this epiphany a lot of our design was able to be solved simply. Instead of having a task of designing two separate systems, our process became that of designing a singular system, and then inverting it for the output stage. Any realizations made while musing over either input or output would benefit the other and vice versa. This massively helped simplify our system at every level, from the main wiring and hardware layout to the software we needed to write in order to run everything.

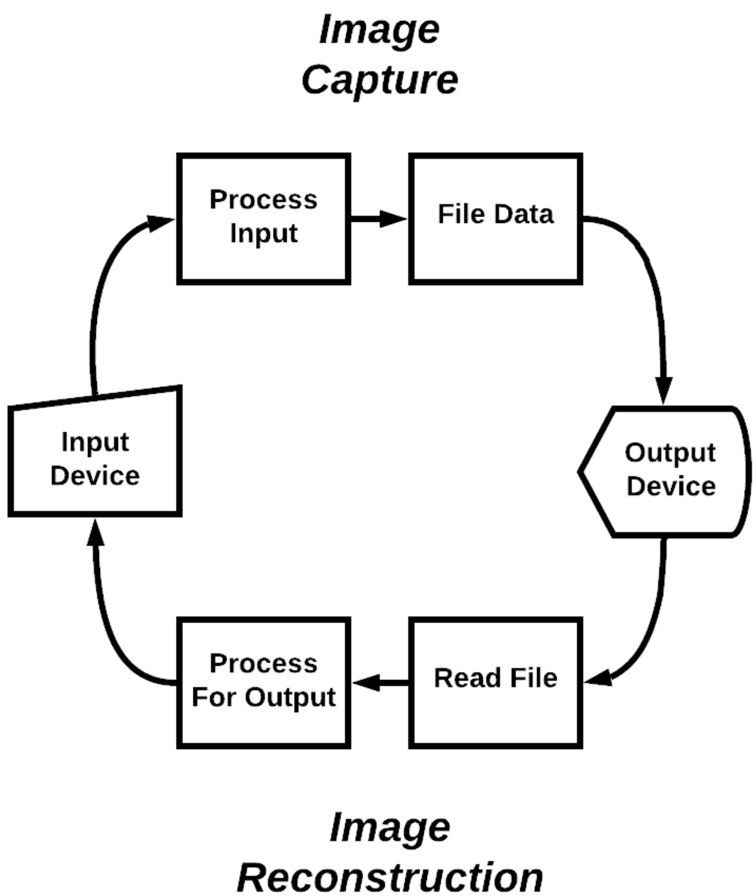


Figure 78: Duality of Image Capturing and Reconstruction

With the new found understanding of the inherent nature of image construction and image capture under our belts we also stumbled upon another helpful tool, albeit one that was far less obvious than the I/O heuristic, the space filling curve. In using the Hilbert Space Filling Curve we were able to drastically reduce the overall complexity of both our software and hardware routing schemes. The ability to compress dimensions of data into a single 1D axis was incredible in its implications for our software layout. We suddenly went from having to make multiple 2D arrays to simple storing all of our information in a single array! Not only was the Hilbert Curve's properties useful for us in regards to software, but it saved us from any troubles regarding hardware routing. We could suddenly quickly assign groups of sensors to be associated with each other and parse them by way of scaling a simple array. Additionally this helped to solve the pesky issue of crosstalk. With the groupings being determined by locality we had a finite methodology for saying what sensors weren't close to each other, and were thus able to group them in a way that would ensure that no neighboring sensors were near each other when sampling. This bonus benefit allowed us to not have to worry about making physical dividers or more convoluted set ups to counter act any apparent crosstalk. Again, as Occam's Razor points out, the simplest path is often times the best path.

17. Administrative Content

17.1 Project Constraints:

- LED Voxel Density
  - Must be dense enough to display a recognizable image.
  - LEDS must not be so close as to totally obscure neighboring voxels.
  - LED shall not exceed 4 inches
- Frame Resolution
  - Ultrasonic Sensors must be arranged close enough to capture a cohesive image.
  - Must be able to display an image in 2D and in 3D.
- Frame Weight
  - Frame shall be light enough as to be able to be mounted on a stand or a wall.
  - Shall not be more than 40 pounds
- Frame Height
  - Shall not be longer than 12 feet
- LED Cube Durability
  - The system shall be able to withstand being deployed several times; joints should be resistant to normal rigors of any standard display.
  - The system shall be able to last for months/years to come.
- Image Quality
  - Image displayed on cube shall simulate fluid motion and be recognizable in form to the objects in front of the Frame.
- Latency
  - System shall be able to process incoming images in a short enough time as to represent a "direct feed" from the Frame unit and out to the Cube display.
- Processing
  - System shall utilize a combination of both analog and software multiplexing schemes to route the data stream.

  - The system shall be able to process the input to a 2D image, and then to a 3D image on the LED Cube

- Cost
  - The cost of the entire build shall not exceed $1000.
- Portability
  - The entire build shall be able to be easily transported.

17.2 Requirements and Specifications:

This section shows the engineering requirements and specifications and the marketing requirements and specifications. It also displays the correlation between each requirement and how we increase one, another requirement may go down. For example, as efficiency goes up, cost will also go up. Or how the portability goes up, the weight will go down. This table is extremely useful because it shows what sacrifices must be made to accomplish what goals and displays it all in an easy-to-read fashion.

↓ = Negative Correlation (i.e. as power consumption is increased, efficiency is decreased)

↑ = Positive Correlation (i.e. as the quality of image increases the cost also increases)

x = No Direct Correlation

Engineering Requirements **Marketing Requirements**

	Frame Rate	Power Consumption	Weight	Cost	Latency
Efficiency	↑	↓	↓	↑	↓
Portability	↓	↓	↓	↑	x
Quality of Image	↑	↑	↑	↑	↑
Size	x	↑	↑	↑	x
Target	30 FPS	Not yet set	< 50 lbs	< $1000	< 60 secs

In addition to displaying both the marketing and engineering requirements, this table also displays the target goals for each of our engineering requirements. These are the goals we set for our project that we hope to accomplish. We want 30 FPS, we want the project to weigh below 50 lbs, we want the project to cost less than $1000 and we want the latency to be below 60 seconds.

17.3 Block Diagram:

The following block diagram demonstrates the general flow of our project and who is assigned what.

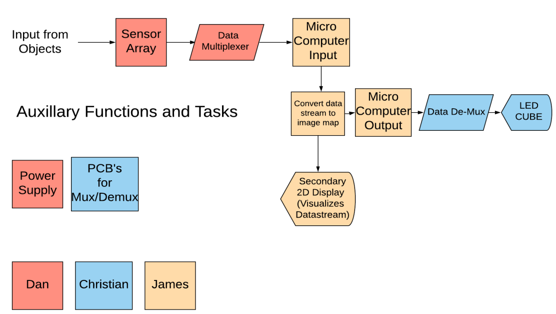


Figure 79: Work Distribution

17.4 Estimated Project Budget:

The following project budget is a general outline of what our project costs will be. The majority of the cost will be in running the LED cube since it consists of the most parts. We plan on staying under $1000.

Range Finder	**$.60 * 64**	**$38.40 Per 8x8 Frame**
LED's	**$0.10 * 512**	**$51.20 Per Frame**
Micro-Computer	**$150 * 1**	**$150**
Spare Electronics for running Cube and Multiplexing the Frames		**$75**
Frame and Hardware for mounting Range Finders		**$10 Per Frame**
Per Frame Cost		**$149**
Total		**$324**

17.5 Project Milestones:

The following timeline is how we plan on moving forward with our project. This timeline is subject to change if need be. However, we will stick to this timeline as best as we can. Timelines like these are important so that you can accurately ensure each small deadline is met. This will keep your project on pace and ready to finish when it is.

Fall 2019:

- **September**: Determine project, write Divide and Conquer V1, meet with professors to get the OK for project design
- **October:** 45 Page Draft
    - 45 Page Draft Meeting
    - Corrections to our 45 Page Paper (if any mistakes)
    - Begin adding more to our 75 Page Paper.
- **November:** 75 Page Submission
    - 75 Page Draft Meeting
    - Corrections to our 90 Page Draft (if any mistakes)

    - Begin adding more to 90 Page Paper
- **December:** Final Submission

Spring 2020:

- **January**: Start working on building the project
    - Order all components needed
    - Build prototypes
    - Test prototypes
- **February:** Testing, debugging project
    - Expand on prototypes
    - Test more
    - Debug project
    - Expand on any features
    - Comprehensive testing to ensure quality control
- **March:** Refining, finishing touches/adjustments/tweaks
    - Tweak Project if needed
    - Refine any details, remove features if needed
    - Ensure Project is ready for demonstration
    - Meet with Professor to ensure project is ready
- **April:** Final Demonstration

By sticking to this timeline, we will ensure our project is able to be completed in a timely manner. If any of these deadlines are not met, we will work diligently to catch back up so that we will be able to meet our deadline

17.6 Looking Forward:

Looking forward, we hope that we will be able to secure funding for this project and able to expand on this project. We hope that this project will be utilized in many different applications either for entertainment or elsewhere. It is our hopes that others may build and expand on our project and add more to it or make projects based off our project. We are excited for the future of this technology in the entertainment field and we look forward to learning more about this field.

There are plenty of benefits to this type of technology. Not only would it be useful in the entertainment industry, it would be beneficial in the medical field. 3D imaging in the medical field is a developing industry and is used for a plethora of medical research, medical practices, and much more.

On top of Medical usage, we could adapt the ultrasonic scanning system to be an extremely powerful form of night vision. By developing further our 2D visualization of the data set and implementing some form of VR goggle support, we could create a night vision system that would be able to work in total darkness. This would be fantastic and especially applicable as it would work around the current limitations of night vision. Most night vision systems leverage the low (but still present) ambient light in a dark space and amplify it by a large amount. This over exposure allows extremely dark objects to become visible. Unfortunately this would not work in a space that was 100% devoid of light. To remedy this, Thermal Imaging is often employed in seriously lowlight situations. Obviously thermally imaging requires that there be relative differences in heat signature between objects in a room so this too as downfalls.

Ultrasonic 3D imaging could be a perfect solution to total darkness sight. As it is working off time of flight via sonics, a Ultrasonic 3D visioning system would be able to nearly perfectly recreate an image in total darkness. Allowing nearly anyone to see a room in total darkness regardless of how hot or cold the objects in it are. This would allow vision of cold areas with 0 heat signatures to be completely visible to any operator.

Our system may need to adapt for a smaller sensor array in size, or potentially a smaller LIDAR based unit if we are to actually adapt the concept to a wearable night vision system. Obvious mounting a 15” X 15” grid of sensors to ones head isn't exactly the most ergonomic solution to seeing in a cold dark (potentially went and rough terrain filled) cave. But as is clear, the underlying technology developed to run Frames as an entertainment unit could be quickly adapted to meet military or medical use cases with some, relative, ease.

Going forward with possible further implementations of this project, we intend on scaling and pitching the Frames unit to various stores and music festivals for installation. Our unit may need to adapt to a LIDAR system to better scale with size and image quality. Additionally, we would scale up the cube to a room size, using a different connection scheme (potentially similar to the installation done by Four Tet). This would be in the interest of durability and scaling as well as overall image quality.

In its current state, Frames is more of a small scale proof of concept of a general idea and technology. With ample budget from a client and time we could quickly scale the device to a room sized installation. This would work well for both stores and festivals. In stores we could position a sensor array in the door way and run the output device in the window. When customers walk through the door they would be picked up on the array and thus displayed on the window. Depending on the client, we could scale the display resolution to be 1 to 1 recreations of the customers or simply display a low res abstraction of the movement of the customers. A similar set up could be installed in a variety of festivals. For an indoor, warehouse style building we could mount the output unit from the ceiling and project incoming audience members or the standing crowd. For an outdoor setting we could simply build a larger standing structure nearly identical to our current LED Cube. This would obviously have to be weatherproofed via wired connections but would be more or less identical in structure and design to our current set up, just larger scale and more power intensive than our current set up.

In short, the scalability and adaptability of the Frames unit is clearly large. Initially conceived as a marketable system for entertainment, the concept as a whole has been developed enough to be easily adaptable to different technologies and the underlying concept has been realized to the point of utilization in different scenarios (eg night vision and 3d imaging through surfaces). We look forward to potentially marketing this device in its current form and believe that it would stand up to many of the major touring installations out in the industry currently.

Without a doubt this project has been an extremely insightful trip through the design and fabrication process that at times taught us more about what it means to be an engineer than much of our collective university experience. We look forward to the start of our collective careers and the beginning of our path.

We currently are preparing a pitch of our device to the IIIPoints music festival in Miami and its partner Space. Our group will be attempting to make a large scale version of the system for an indoor concert setup. In this specific pitch we are seriously considering using a LIDAR system to lower the overall cost of the unit. Unless we can get a very large budget for the project it'd be better for us to spend the majority of our funding on output systems than the input scheme. As this would increase the scale and effect of the device system as a whole due to it being a primary entertainment design whose goal is to be seen “in passing”. Effectively we do not need to worry to much about image clarity unless a potential client is willing to support the group with requisite funds to actually build and design a high resolution 3D imaging system.

18 Appendices:

18.1 References:

- "What Is USB (Universal Serial Bus)?" Computer Hope, 4 May 2019, https://www.computerhope.com/jargon/u/usb.htm.
- Alarco, Jose, and Peter Talbot. "The History and Development of Batteries." Phys.org, Phys.org, 30 Apr. 2015, https://phys.org/news/2015-04-history-batteries.html.
- "USB History." Premium USB, https://www.premiumusb.com/usb-history.
- "First US Detachable Electric Plug – Today in History: November 8: Connecticut History: a CTHumanities Project." Connecticut History | a CTHumanities Project, https://connecticuthistory.org/first-us-detachable-electric-plug-today-in-history-november-8/.
- FutureLearn. "Persistence of Vision: How Does Animation Work?" FutureLearn, https://www.futurelearn.com/courses/explore-animation/0/steps/12222.
- Instructables. "LED Cube 8x8x8." Instructables, Instructables, 26 May 2019, https://www.instructables.com/id/Led-Cube-8x8x8/.
- "ISO/IEC 14882:2003." ISO, 1 Sept. 2011, https://www.iso.org/standard/38110.html.
- ISO, https://www.iso.org/obp/ui/#iso:std:iso:12224:-1:ed-1:v1:en.
- Anurag. "Raspberry Pi vs Arduino vs Beagle Board." Cloud Computing, https://www.newgenapps.com/blog/raspberry-pi-vs-arduino-vs-beagle-board.
- *ISO*, https://www.iso.org/obp/ui/#iso:std:iso:12224:-1:ed-1:v1:en.
- "ISO/IEC 7816-12:2005." *ISO*, 5 Dec. 2018, https://www.iso.org/standard/40604.html.
- "Multiplexed Display - the Basics." *LEDnique*, http://lednique.com/display-technology/multiplexed-display/.
- FutureLearn. "Persistence of Vision: How Does Animation Work?" *FutureLearn*, https://www.futurelearn.com/courses/explore-animation/0/steps/12222.
- Smith, John SmithJohn, and John Smith. "3D Hologram and the Future of Technology and Entertainment." *Business 2 Community*, https://www.business2community.com/tech-gadgets/3d-hologram-future-technology-entertainment-01214803.
- Weber, Marcus. "How Does a 3D Hologram Work? - Meee Services: Electronics Design." *MEEE*, 4 Dec. 2018, https://www.meee-services.com/how-does-a-3d-hologram-work/.
- Klien, Asher. "Raspberry Pi vs. Arduino Vs. BeagleBone – What's The Difference?" *Vilros.com*, Vilros.com, 17 Jan. 2019, https://vilros.com/blogs/getting-started/raspberry-pi-vs-arduino-vs-beaglebone-what-s-the-difference.
- Aggarwal, Param. "Mapping A Room Wh Ultrasonic Distance Sensors." *Hackster.io*, 7 Jan. 2018, https://www.hackster.io/paramaggarwal/mapping-a-room-wh-ultrasonic-distance-sensors-9725b7.
- *YouTube*, YouTube, https://www.youtube.com/watch?v=z8ntr5lYxGU.

- Instructables. “2D Ultrasonic Mapping.” *Instructables*, Instructables, 3 Oct. 2017, https://www.instructables.com/id/2D-Ultrasonic-mapping/.
- “First US Detachable Electric Plug – Today in History: November 8: Connecticut History: a CTHumanities Project.” *Connecticut History | a CTHumanities Project*, https://connecticuthistory.org/first-us-detachable-electric-plug-today-in-history-november-8/.
- Alarco, Jose, and Peter Talbot. “The History and Development of Batteries.” *Phys.org*, Phys.org, 30 Apr. 2015, https://phys.org/news/2015-04-history-batteries.html.
- “What Is USB (Universal Serial Bus)?” *Computer Hope*, 16 Nov. 2019, https://www.computerhope.com/jargon/u/usb.htm.
- “Political Constraints.” *Consequential LCA*, https://consequential-lca.org/clca/marginal-suppliers/political-constraints/.
- “Engineering Safety Requirements, Safety Constraints, and Safety-Critical Requirements.” *The Journal of Object Technology*, http://www.jot.fm/issues/issue_2004_03/column3/.
- Google. “Google/Hilbert.” GitHub, 22 Nov. 2018, https://github.com/google/hilbert?fbclid=IwAR2j5-JaX_fMjesdWXo44TPAM4SGkEKX9SgRvTMYrnFkcAaLbYlaF2cWo0A.
- Ruohan, Gao. *2.5D Visual Sound*. CVPR, http://openaccess.thecvf.com/content_CVPR_2019/papers/Gao_2.5D_Visual_Sound_CVPR_2019_paper.pdf.

18.2 Permissions:

CM

Christian Moen <chrsyntaks@gmail.com>
Fri 11/15/2019 8:05 AM
Daniel Hureira

Hi,

You may use any pictures of my LED cube from instructables.
It would be fun to get a copy of your report when it is finished :)

Best regards
Christian Moen

Will do. | Will do, thanks! | Good idea!

Are the suggestions above helpful? Yes No