# HoloMon :
# UCF Senior Design 2 Document

EEL4915 Group A, Lei Wei, Fall 2022

Nathaniel Kissoon, Cp.E.

Elizabeth Mikulas, E.E.

Diego Rodrigues, Cp.E.

Taniya Shaffer, Cp.E.

# Table of Contents

# 1. Executive Summary

This project is tasked at ideally rendering a 3D hologram inside a display surface that can interact with a user via gesture based recognition. Through using several, interconnected subsystems, the team will attempt to build this project. The first is a holographic display, which will not require any external interfaces like glasses to view, but act as a standalone device. The second hardware subsystem is a microcontroller unit that distributes power to the third and last system while using a motion sensor to turn on/off the webcam. The last hardware system is a webcam programmed with a trained ML program that recognizes hand gestures and has the hologram react accordingly. The interconnecting software application, which will be a mobile application, will allow for user-hologram interactivity without necessarily putting too much strain on the actual 3D display itself.

The premise of these interconnected subsystems is to provide a user experience that is natural, defined as not needing extraneous devices like smart glasses, to interact with a virtual environment. The applications of this project will be focused in mainly entertainment and design, with some potential use cases being in accessibility.

Despite some features of the requirements already existing, the combination of these major subsystems is a novel idea we will call “HoloMon.” We have received copyright permission from The Pokémon Company to use their assets so long as we are not profiting off their intellectual property. Therefore, when referring to the actual asset, we will call it “Pokémon” but whenever we refer to the device or project, we will call it “HoloMon.” Our overarching team objective is to provide a proof-of-concept design and prototype that will fit within our current budget (see 8.3 Project Budget and Financing). Another important team objective is to exercise and demonstrate our individual and collective abilities to work in a team for a project that requires expertise in electrical engineering, computer engineering, and computer science skill sets. In section 8, we set a projected timeline for the lifespan of our project and in section 4.2 we list out constraints that will heavily influence the design of our project.

The influencing factors for HoloMon’s ultimate design and creation will be the research and limitations of our selected system software, software APIs, hardware systems, and hardware components. Some of the technologies we will be implementing include an ML model, mobile application, gesture recognition, serial communication protocols, PCB design and manufacturing, microcontroller connections, server API, and Pepper’s Ghost Illusion. For the most part, “real” holographic systems as well as volumetric 3D displays are currently under development and the technology for them is hidden behind proprietary corporate IP. Given this fact, our proof-of-concept will be made with less than ideal technologies that are within our budget. However, we still hope to see HoloMon as a potential application of future entertainment systems and other appropriately applicable systems that provide a more natural interaction between humans and computers.

# 2. Project Description

In this section, project description, the following qualities subsections describe the project motivation goals, objectives, requirements specifications, and quality of house analysis. In the project motivation goals, the categories are divided into base goals, advanced goals, stretch goals, and functions. In requirements specifications, it is primarily separated into two tables that define criteria and requirements.

## *2.1 Project Motivation and Goals*

Holograms have not been the most simplistic invention of all electronics. It was first sparked as an idea in science fiction. One of the most popular Holograms is known in movies such as Star Wars. Many futuristic films or tv shows have shown people using a phone that had holographic features that could be used for phone calls. Once this idea of creating a 3D like image using light and color came into light, engineers have been working to create an accurate Holographic image to depict images.

The first Hologram was created in the mid 1900's and was introduced through lasers. There are many ways to go about creating a realistic Hologram such as utilizing lasers, phones, and or oscillating LED's.

There are a couple disadvantages with Holographic images depending on which route is chosen. If the engineer decides to utilize 3-D goggles to create the Holographic effect, this route is very expensive. Not only is it expensive, but it is very difficult to project an image in 3-D with light. Another route when creating an image is using lasers. To do this, the engineer must have a record of interference patterns that occur from the laser meeting its own light when the object is lighting up. This route is less expensive than 3-D goggles, but the intricate details give the invention a very small door for succeeding. It takes a lot of time and has very little room for error. Oscillating LED's are less complicated to create and less expensive as well. This is what will be used when creating the HoloMon. The disadvantage of using the FlexLED is that once it is created, its image is not easily seen from every POV. The only way a FlexLED can be seen is if the user is standing directly in front of it and on the same level. However, even with this disadvantage, the advantages outweigh this one constraint.

There have not been many Holographic toys due to the fact that Holograms are not easy to create. There have not been many accurate Holographic merchandise in the field due to the complex designs. However, if one was done correctly, this would spark a large opportunity for the Holographic realm.

The COVID-19 pandemic has significantly changed social interactions. From social distancing to quarantining, it has been a very confusing time for almost everyone. This is especially true for children between the ages of 4-7 years old, where they grew up without the regular social interactions we are accustomed to.

HoloMon is a solution to this unforeseen collateral on part of the pandemic. We want to create an interactive robotic hologram that will provide a comfortable and cool way for children to acquire social interactions with their friends and peers. Additionally, playing Pokémon Shining Pearl during quarantine and watching a favorite Pokémon follow your character around in game and interacting with other NPCs was the inspiration that sparked the idea of this project.

## *2.2 Objectives*

To create an interactive robot to provide a surreal experience for kids. As a team, we want to have a good balance of software and hardware challenges.

### *2.2.1 Base Goals*

*-The priority of this project is to create a hologram:*

This hologram will be looking like a specific character from the HoloMon drawings. The Hologram will be a box-like structure that will project a figure in a darker environment.The amount of characters for the child to pick from has not been decided, but probably close to 4-6 different characters. In regard for the basic goal, we will say there will be only 1 character. This character will be projected through the hologram once the user turns on the TV. Some sort of glass most likely plexiglass will be used to create a projection of a character. Mirrors may be utilized as well. The base goal of this project is to have a stationary Hologram that will interact with the user. This hologram will most likely fit best on desks as it will be a heavy box-like structure.

*-The hologram will only be visible at one spot:*

The user will not be able to look at the HoloMon directly unless they are pointed facing directly towards the HoloMon. Fortunately if the base goal of the HoloMon is achieved, this won't be an issue since most likely the HoloMon will be interacted with the user face-face on a desk.

*-The hologram will be able to interact with the user.*

The complexity of the project lies within the communication of electronic devices and the user to get the HoloMon to interact with the User. Most likely there will be a camera and some sort of communication from a computer based board that will send data to an application/mobile device and move the HoloMon. The cloud base will be used to create the HoloMon character who will move depending on what gesture the user makes. Most likely the communication between the HoloMon and the user will be hand gesture based such as a wave in the x or y

direction. There will be instructions for the user with gestures to use with their HoloMon and how to communicate with them. The HoloMon will be projected on a TV screen and will be projected towards the user.

### *2.2.2 Advanced Goals*

*- Allowing the user to choose 2 characters that will appear on the hologram:*

The two characters will be designed by the team members and projected through the hologram. The HoloMan will have a very distinct and animal-like presentation.

*- Ability to change hologram via phone application:*

The user of the HoloMon will be able to choose 1 out of 2 characters they would like to have follow them. The selected Holoman will be their buddy that they can interact with. If the user wants to select another buddy, they must select the character again using the mobile app.

*-The HoloMon will be able to interact with its user via 1 or 2 hand gesture(via Hologram):*

All gestures will be recognized through a webcam installed on the robot. The most basic feature for the robot is when the user waves to the HoloMon, the hologram will wave back. There is a whole realm of gesture recognition. One more idea is that when you clap the HoloMon will clap back.

*-The robot will have protective gear surrounding the robot/HoloMon.*

Doing so will ensure that any collisions will be mitigated upon impact.

*- Robots will have collision detection due to sensors around the base of the robot:*

This is an advanced feature because the robot will be following the human. If the human does so correctly, they can help their robot avoid obstacles such as rock and or small bumps. However, the accuracy of the human helping the robot avoid all obstacles is most likely not good since the main target audience is children. When the robot detects something in its path it will stop until the user removes the object from its sight.

*-The Hologram will be adjustable by the user utilizing mirrors that will be moved until the user can see the HoloMon:*

As labeled in the constraints, the way the Hologram is being created, the user will be unable to see the Flexed since it is only facing one direction, straight. However, if one mirror is placed in front of the HoloMon and one behind, it will create a view for the user in front of the HoloMon.

### *2.2.3 Stretch Goals*

-*To create a robot that will follow a human hand and or body around*:

This will be completed through sensors such as infrared and ultrasonic. The infrared sensor will detect an object in respect to an angle while the ultrasonic will detect an object through long distances. Together they will define an accurate distance detecting robot. The Arduino Uno will be the base of communication for the sensors and the motors.

*-Have 3 gestures created by the HoloMon will be via Hologram (not robot):*

Creating gestures using Flexed will be complex and difficult to communicate. However, not impossible. All gestures will be created through the visuals of the Hologram.

*- User can choose between 4 HoloMon's via mobile device:*

Four HoloMon will be selected by the user through a mobile app. The designs will all be created through drawings created by the team.

*- Robots will have collision detection and will be able to go around the obstacle and continue to follow the user again:*

This will be a little complex because the robot will not only have to distinguish the difference between a human and a random object, but will have to move around the object and then follow the human hand again.

### *2.2.4 Functions*

1) Users will be able to pick out which HoloMon they want through an app.

2) Users will be able to see the HoloMon in a hologram.

3) Users will be able to interact with HoloMon (such as waving).

## *2.3 Requirements Specifications*

The following is a list of criterias set forward to produce a high-quality and cost efficient product. This is a list of high level criterias that qualify the product as desirable as possible for the end user. This list will then be used to publish the technical requirements and specifications of the final product.

| Criteria Number | Criteria Description |
|---|---|
| 1.1 | The user-HoloMon interaction will be based on the “rangefinder” motion sensor. I.e. if the range is <2m, the pond accordingly. |
| 1.2 | The user-HoloMon interaction will also be based on a ML model and webcam for user-HoloMon interaction. |
| 1.2 | The user-HoloMon will create an account and choose the monster to be displayed as a hologram through a mobile application. |
| 1.3 | Users shall be able to update their data(credentials, settings, etc.) as needed through the mobile client. |
| 1.4 | User data shall be stored in our web server (Azure, AWS, etc.). |
| 1.5 | There shall be a variety of pixel art HoloMons to choose from (At least 3). |
| 1.6 | Each distinct HoloMon shall have at least 3 animation assets to give the user interactivity. |
| 1.7 | API shall be concurrent, allowing multiple users to interact with our database. |

**Table 1. *Criteria***

The following is a list of technical specifications determined to the product’s criteria while establishing a high-quality design. These technical requirements will be used as a guide of development of the project. The highlighted requirement numbers are the minimum-viable product (MVP) that the team will showcase at the end of senior design 2.

| Requirement Number | Requirement Description |
|---|---|
| 1.1 | Stationary hologram of 30”x30”x50” that displays the HoloMon using a device. |
| 1.2 | Ability to change the HoloMon in the mobile app with 3 different characters. |
| 1.3 | User-HoloMon interaction is through gesture-based recognition driven by a camera and machine learning. |
| 1.4 | HoloMon devices should stay at least one hour active without the need to recharge. |
| 1.5 | 5 GB AWS MongoDB Database to hold enough user information. |
| 1.6 | HoloMon should have a maximum of one second delay to respond to the user’s gestures. |
| 1.8 | Cost to be less than $400. |

**Table 2. *Requirements***

The next figure below is the legend for work distribution on how the team will construct the project in the next semester is outlined as the first 4 boxes with their names. Anything red will be done or led by Diego. Similarly, anything in yellow will be done or led by Nathan. For Elizabeth, anything in blue will be done or led by her. Finally, anything in purple will be done or led by Taniya. By following this legend, we can ensure that each of the team members has an important part to contribute within this project.


**Figure 1. *Work Distribution Legend***

Beneath this legend, Figure 2, outlines the project software block diagram and how the team will construct the software side of HoloMon next semester.

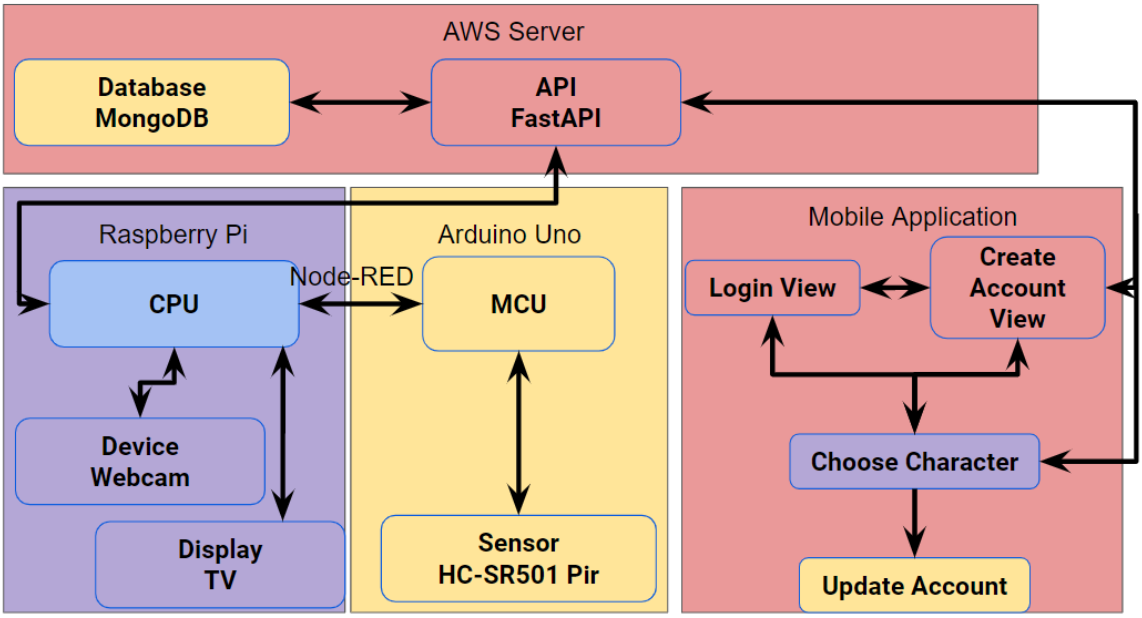


**Figure 2.** ***Project Software Block Diagram***

The next diagram, Figure 3, displays the project hardware block diagram and how the team will commit to creating the hardware systems next semester. The red outline symbolizes the hardware side of components, whereas the blue outline in the previous diagram symbolized the software side.

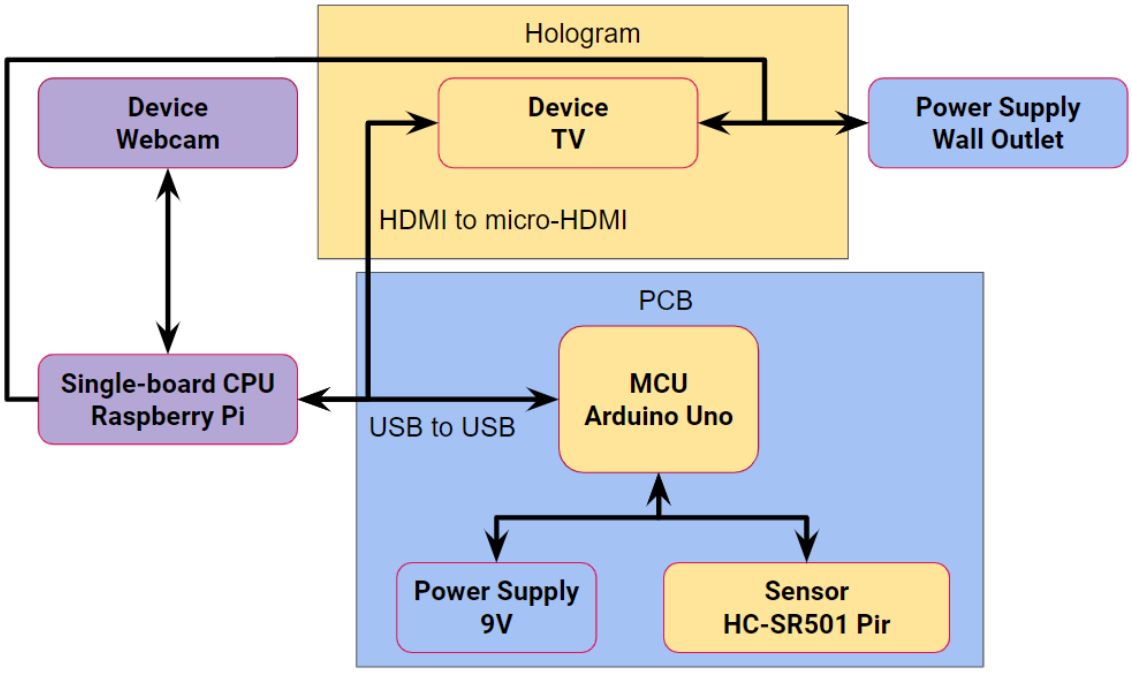


**Figure 3.** ***Project Hardware Block Diagram***

## *2.4 Quality of House Analysis*

We can analyze trade offs between various qualities in the technical specifications and constraints using the criteria and specifications presented in the preceding subsections. This can be arranged into a quality house to visualize and focus on the most critical components of the design that must be considered.

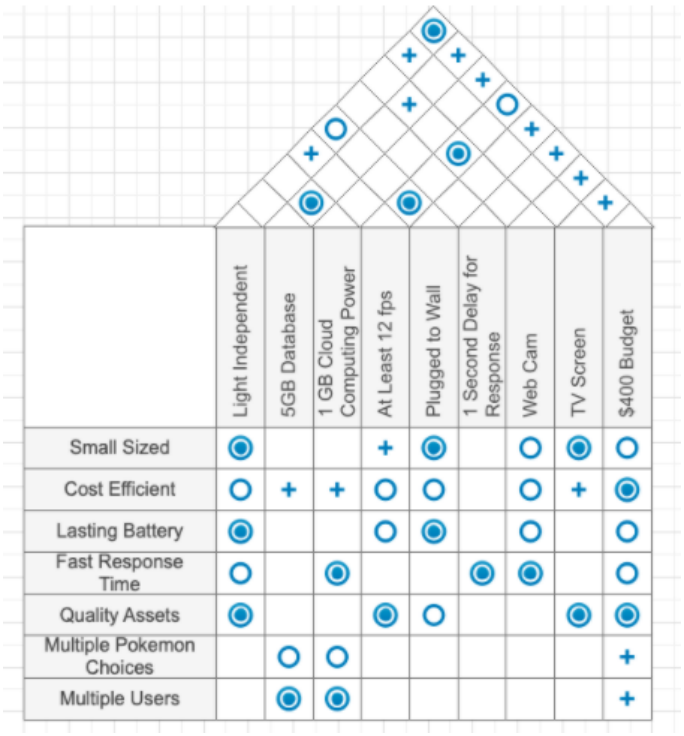


**Figure 4.** ***House of Quality Diagram***


**Figure 5.** ***House of Quality Diagram's Legend***

# 3. Research related to Project Definition

In this section, the team broke down the necessary components in their design to create HoloMon and the research to make that happen. Research related to project definition includes subsections named: existing similar projects and products, relevant technologies, strategic components and part selections, possible architectures and related diagrams, and parts selection summary. For clarity, the research conducted on each component has a comparison table at the end of the subsection. The yellow highlighted component is the one selected and will be summarized in the parts selection summary. Further justification of the selected part will be given in the strategic components and part selections subsection.

## *3.1 Existing Similar Projects and Products*

This subsection, existing similar projects and products, focuses on other products that currently exist to show a more thorough explanation of the design and concept of HoloMon as well as act as inspiration for the team. The main inspiration behind creating HoloMon had to do with the Pokémon games but one of the features the team wants to see implemented in real life is the ability to interact with the chosen Pokémon like in Pokémon Go.

### *3.1.1 Pokémon GO / Pokémon Let's Go*

Pokémon GO is a free-to-play mobile app for Android and iOS devices released in 2016 and developed by Niantic. The main objective of the game is for players to catch pokémon in the wild by walking around while the app uses the phone's GPS to locate pokémon. There are other locations to explore in the game like PokéStops and Gyms that give the players to complete extra objectives. The inspiration HoloMon pulled from this game is the idea that the user can interact with pokémon in a more extensive way than in the base games. For example, being able to pet them just by rubbing a finger over their head, playing catch with them, and feeding them.

While it was difficult to procure an image of the interaction between the user and the pokémon from Pokémon GO, there was a similar game released called Pokémon Let's Go Pikachu / Eevee. In this a user can interact with a pikachu or eevee and shows the same kind of interaction that the team wants to generate for HoloMon. By hovering over the creature, the user can "pet" the pokémon and do other interactions with it as seen in figure 6 from [36]. Many of the base games only let you feed the pokémon or have one follow you around at a time.

**Figure 6.** ***Pokémon Let's Go Eevee Petting***

### *3.1.2 Olomagic*

Olomagic is a company creating holographic displays that generate 3-D images in mid air. There are different versions of housing the holographic units that proved as inspiration for how HoloMon would also be displayed.

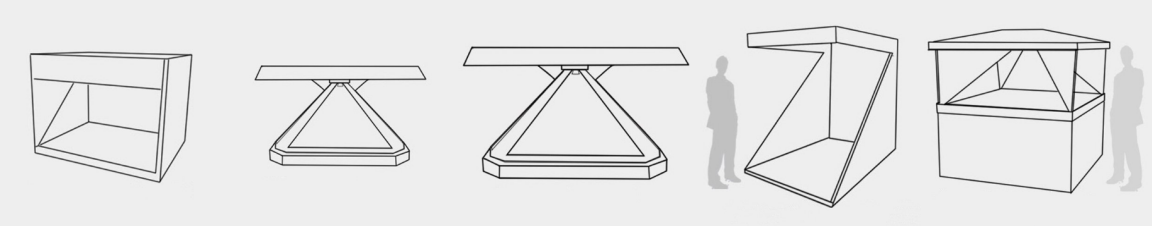

**Figure 7.** ***OloMagic Products***

Their main functionality is to provide a new way of "marketing and communication that conveys an impressive message to the customer or spectator," [30]. Their targeted audience includes retail establishments, conferences, shopping centers, museums, airports, and events. An important aspect of including Olomagic in this section is that each of their products listed technical specifications often not found in other projects and products.

For example, the Olomagic 90 Degrees is one of their products that offers a 90 degree front view while being able to project a larger hologram than other pyramidal models.

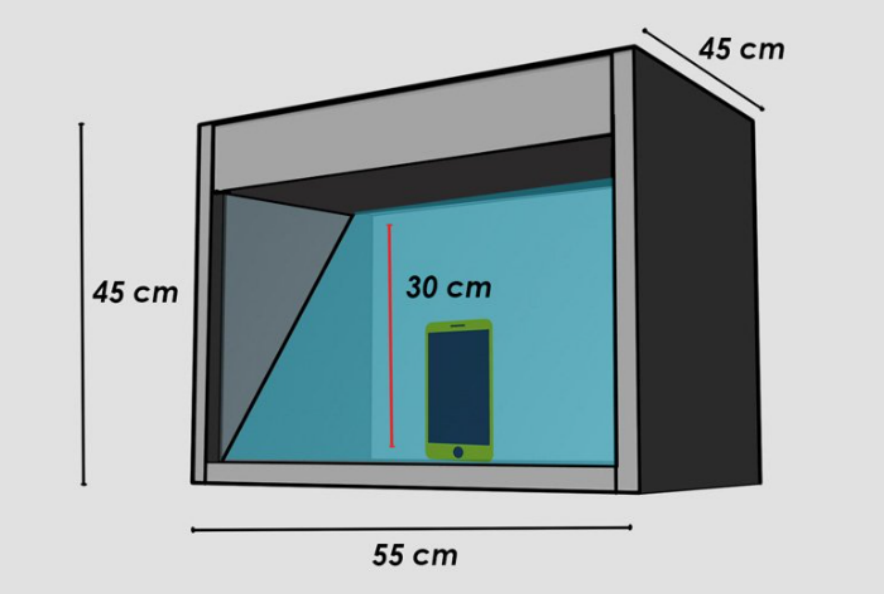


**Figure 8.** ***Olomagic 90 Degrees Dimensions***

The characteristics of the Olomagic 90 Degrees beyond the initial statement is its resolution in full HD while having a screen size of 19”, the possibility of combining physical products with holographic images, and that the content can be replaced with a USB harddrive that supports WWV, MP4, AVI, MPEG, and H.264 [30]. On top of the actual hologram itself, the interior has adjustable illumination LED colors by RGB, audio speakers, lacquered wood, and HDMI input. It was particularly important to note that the Olomagic 90 Degrees requires a power input of 230 V at 50-60Hz as it informed the team of the kind of power supply necessary to drive HoloMon’s design decisions. The price for the Olomagic 90 Degrees is approximately $1365.36 converted from 1250 Euros.

### *3.1.3 Holographic Audio Visualizer with Motion Control*

The team had already decided to have some kind of gesture recognition approach be part of HoloMon. However, this project, the Holographic Audio Visualizer with Motion Control inspired some ideas around how to physically implement the camera-based gesture recognition platform for HoloMon. The function of this project is exactly what it’s named after and shows real time animations of SoundCloud playlists. In their list of hardware components for the project, they utilize Pi Supply Flick Large, Raspberry Pi 3 Model B, a computer monitor, acrylic sheet, black foam PVC sheet, wood, countersunk wood screws, socket cap screws, nuts, and spray paint.

How this works is that the display (computer monitor) projects an image down onto an acrylic pyramid which creates a 3D effect using Pepper’s Ghost Illusion (subsection 3.2.1.5 explains this technology in further detail). The display has audio visualizers inside which respond in real time to music and can be

controlled via a gesture control interface board. This board can detect various hand motions without being touched. Simply swiping or turning a hand in the air above the board will allow the user to play the next song, pause, adjust the volume, switch the visualizer, and other functions.

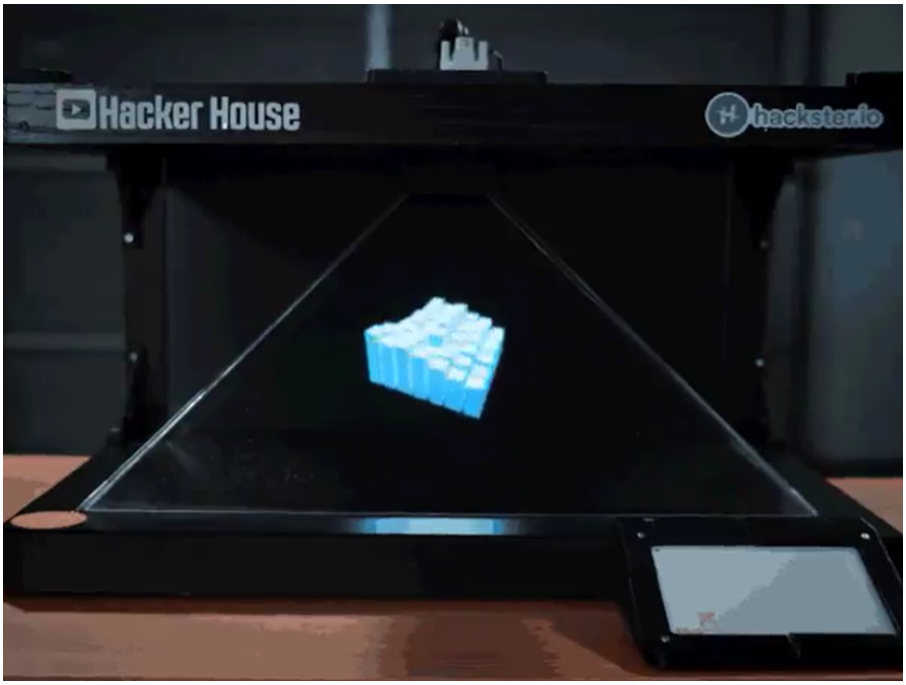


**Figure 9.** ***Holographic Audio Visualizer with Motion Control***

Above is an image of the project that shows the gesture control interface board is a part of the housing unit that holds the monitor and the pyramidal acrylic glass.

3.1.4 SolidLight™ by Light Field Lab

Light Field Lab is a Silicon Valley startup founded in 2017 with one mission: to design the world's most innovative holographic display ecosystem [34]. Their primary work, SolidLight™, is not exactly a hologram but rather what they consider to be the next generation of display that combines "unprecedented resolution and density to accurately project dimensional wavefronts to form objects that escape the screen and merge with reality" [34]. In comparison to the usual 3D projection of 2D images, SolidLight™ brings a real hologram experience that accurately moves, refracts, and reflects in physical space.

This is one of the many examples of modern holographic technology that would be ideal to implement for HoloMon, but ultimately was too expensive and the technology is too new for all the constraints later mentioned in section 4. The main thing that was useful from viewing Light Field Lab's product is figuring out how much of a hologram it would be possible to make on the team's budget. This influenced the design decisions to make something more simple, since the original concept was to have a viewing angle of 360 degrees of the hologram without glasses or external equipment. However, the team remains hopeful that

HoloMon will someday utilize technology like SolidLight™ to be used as a mobile entertainment system.

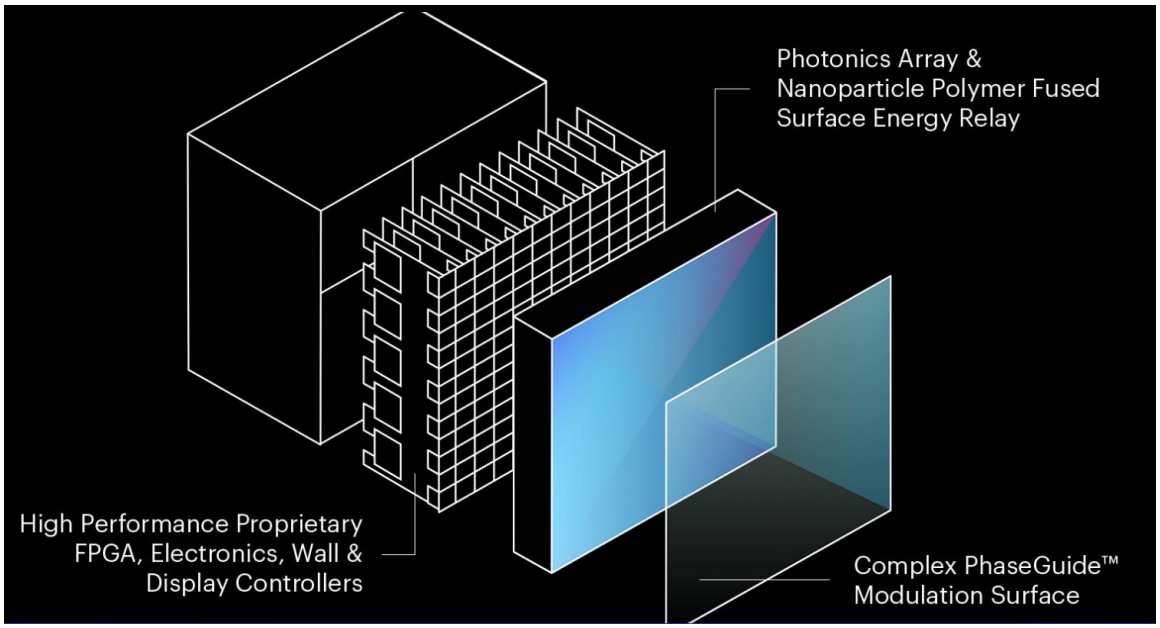


**Figure 10. Light Field Lab's SolidLight™ Technology**

The technical specifications of SolidLight™ is the display architecture is a 28" (.7m) modular self-emissive – SolidLight Surface panels, the type is a broad-spectrum complex-amplitude dense converging wavefront, the sample density of the display is 10 billion pixels / m2, the image format is a wide-gamut 10bit HDR at 60 Hz, and the software and rending is a proprietary program call WaveTracer™ plugins and factory calibrations LUTs [34].

3.1.5 Pokémon Holograms with Image Tracking

This project was done by a YouTuber named KennyWdev and the technical specifications are unknown [35]. However, it did serve as one of the main projects that the team looked at for inspiration to see how to get started with creating a 3D projection of a 2D image. In this case, there is most likely a scanner that can read QR codes and special QR cards at the bottom that have predefined GIFs on them. It is then read by the scanner and the proper image is projected onto the display which is then refracted using the common pyramidal acrylic glass technique (see subsection 3.2.1.5) to create the illusion of a 3D image.

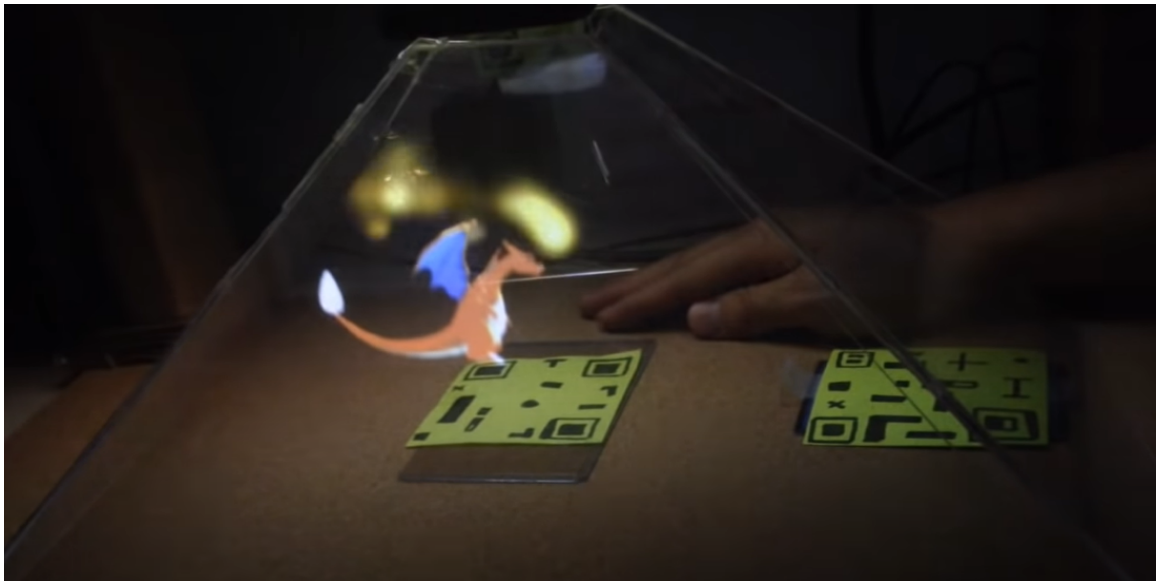

**Figure 11. Pokémon Holograms with Image Tracking Demo**

This was one of the many projects using this popular technique and therefore became a very influencing design factor since it was the most commonplace. Many projects claiming to be holograms used the pyramidal acrylic glass approach and found very little evidence to prove that there is an otherwise similarly cheap alternative to projecting a 3D image without the same drawbacks.

## *3.2 Relevant Technologies*

In the following subsection we will discuss the research that took place as we strived to find the best overall design of our product, compartmentalized into the major design components of HoloMon.

### *3.2.1 Hologram / 3D Display*

#### *3.2.1.1 VR*

The Covid-19 pandemic has dramatically changed many people's way of lives, and this is not going unnoticed by tech corporations looking to capitalize on future technological trends. The buzzword, "metaverse", has become a hot topic in 2021 and going into 2022 with many corporations investing heavily into it. Most notably to embrace this trend is Facebook, now rebranded as Meta. They have created what is currently the most popular VR headset series, with their newest version being the Oculus Quest 2. The Oculus features a VR headset that has a display split between the eyes allowing each eye to be shown a different feed creating a stereoscopic 3D effect. It also tracks your position in space and orients

your point of view to mimic in the virtual world. The complimentary controllers use a visual tracking system called Oculus Insight that uses infrared LEDs in the controllers.

**Figure 12. *Oculus Quest 2***

While VR is a hot topic right now, the difference between VR and holograms is that holograms can be seen with the naked eye as it portrays the image in the real-world. Using VR would incur high overhead costs such as a headset and would differentiate the user from the real-world which we don't want. At the time of writing, the cost of the latest version is $299.

### *3.2.1.2 AR*

Augmented Reality (AR) differs from virtual reality because it overlays virtual features onto the real-world instead of completely reconstructing it. An AR system can be defined to consist of 3 features: real and virtual objects, real-time interaction, and accurate 3D registration of objects. Perhaps the most popular AR headset in recent times is Microsoft's HoloLens. Much like the Oculus, the HoloLens requires a bulky headset which doesn't match our needs. It is also important to note that the HoloLens have been discontinued by Microsoft as they explore other mixed reality options with Samsung.

**Figure 13.** ***Microsoft HoloLens***

Mobile AR has also made an emergence in the past couple of years, where the only requirement would be a smartphone. Unlike the HoloLens, the user would use a smartphone through which they would see the virtual objects. However, this would not be suitable for our product as we would want it to be seen with the naked eye without the need for any additional hardware. The cost of the Hololens 2 is $3500.

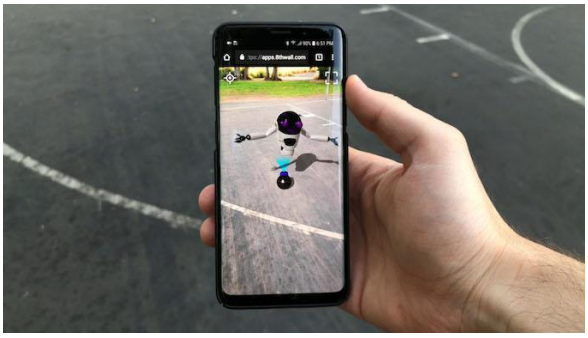

**Figure 14.** ***Mobile AR***

### *3.2.1.3 LED-Based Hologram*

Recently a popular trend in the hologram field is to take an LED-based approach. This is because it is fast, simple, and relatively cheap to set up. It is being used in many areas from drone displays to advertisements and casual projects. It is important to note that drone holographic displays currently operate mainly on a large scale, so we will not consider that possibility. LED holograms come in many forms, but they operate through a similar principle. Probably the most common form is LED fans. They produce an illusion of a 3D object floating in space. The fan is spinning so fast that it becomes invisible to the naked eye and becomes see through. The LEDs are lit up in such a way while the fan is spinning that it tricks the human mind into thinking it is a full image.

**Figure 15.** LED Holographic fan

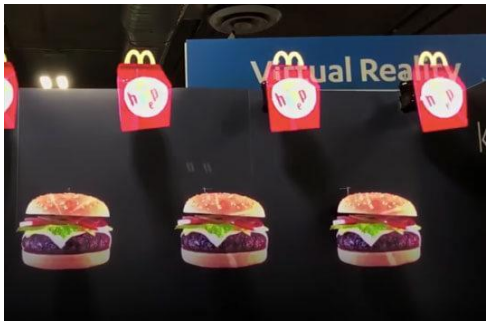

**Figure 16.** Advertisement using holographic fan

Some limitations of this technology are that it is not a true hologram in the sense that it does not produce a 3D volumetric image but just an illusion that can be seen from the front of the display. There are also safety concerns with this because it would need encapsulation due to the

spinning fans or be placed out of reach. In addition, the fans produce noise and generate a significant amount of heat, so immersion and power management become significant issues. The price is ~$100.

### *3.2.1.4 Parallax Barrier*

Parallax barrier is a device placed in front of an image source, such as an LCD which creates a stereoscopic viewing experience. This stereoscopic view is created without the need for 3D glasses or any other sort of eye wear. The device itself is opaque with precisely positioned slits that allow the eye to see a different set of pixels which in turn create the parallax effect.

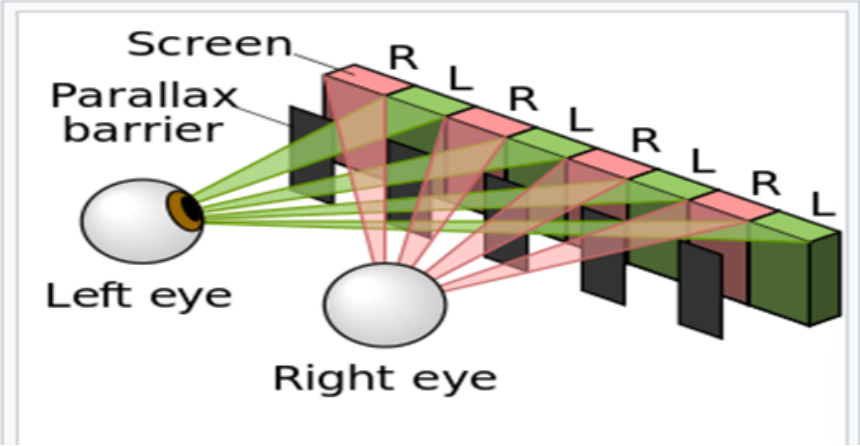


**Figure 17.** *Parallax Barrier*

This is a quite simplistic approach and is not without its shortcomings. The viewing range of a screen using this device would be quite limited and would not be ideal for use in big screens like televisions. The 3D effect would only be limited to a certain range. To resolve this problem face-tracking technology is being used to track the position of the viewer and adjust the positioning of the slits accordingly. Another disadvantage of using a parallax barrier is that the resolution would decrease. More specifically, because the horizontal pixel count in each eye is halved, the overall horizontal resolution of the image is halved. Looking at the screen for too long may also be nauseating and hurt the eyes.

The first known commercialization of the parallax barrier was in the early 1900s. In the 2000s, Sharp introduced two laptops with 3D LCD screens. As of now Sharp has discontinued their production but other companies still are working on them. In the Japanese market, Hitachi developed the first 3D mobile phone. One of the most notable commercial uses of a parallax barrier was the release of the Nintendo 3DS. The cost is $5.

### *3.2.1.5 Pepper's Ghost Illusion*

The pepper's ghost illusion is a technique where an image is reflected on a transparent screen at a 45-degree angle, viewers will see a virtual image with depth in the middle of the air.

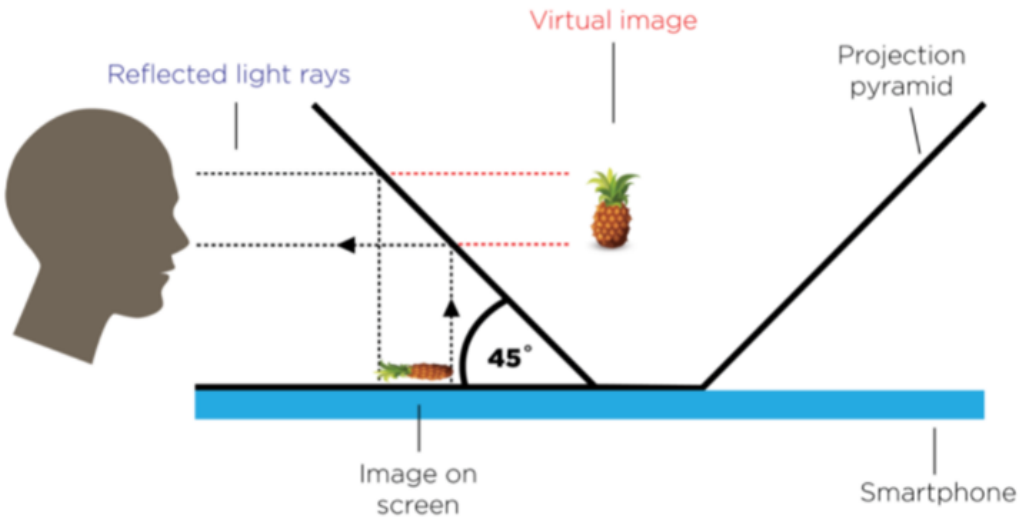


**Figure 18.** ***Pepper's Ghost Demonstration***

Pepper's ghost illusion is a technique that originated in the 1800s by an English scientist named John Henry Pepper. It has become popular in cinemas, amusement parks, museums, and concerts. This technique can come in many sizes. In amusement park attractions and concerts, whole rooms can be dedicated to achieving a life size scale of an image or video. In contrast, small scale projects can be achieved using a small phone or tablet screen. For our project, this is most likely the size that we would use.

Although the pepper's ghost illusion is often mistaken for a hologram it is by definition not a hologram. A hologram is essentially a three-dimensional photograph of an object. It is created by shining a laser at the object from different angles and recording it onto a holographic film. The produced hologram will be 3-D and as your perspective of the image changes, so does the image itself. Laser holograms will be talked about more in the following section.

In contrast, pepper's ghost hologram does not change as the viewing perspective changes. This means that it is not an autostereoscopic view. There will also be crude viewing angles as it will be projected in a transparent pyramid structure with corners where viewing isn't possible. Scaling the projected image to the desired dimensions will also be challenging and will mostly likely be a case of trial and error.

While the pepper's ghost illusion is not a true hologram, it has many benefits. For one it is significantly cheaper than any of the other options and easier to implement. Given our tight budget and the exorbitant cost of the other solutions this may be a viable choice. The cost would be the price of an LCD and Plexiglass. Given our group already has an LCD display the cost would be just the price of PlexiGlass which is around $20. If another LCD were to be used the price would range on the size of the display from tens to a couple hundreds of dollars.

#### *3.2.1.6 Projectors / Laser Holograms*

In laser holography, a hologram is recorded using a laser light as its source. Laser light is an extremely important element in making holograms as opposed to normal white light. Unlike white light, laser light moves all in the same direction, has the same wavelength, and all the light waves are in phase. When two coherent laser beams meet, they create striped interference patterns. When the reference beam and object beam meet, they create a special interference pattern that encodes 3D information of the object on the holographic film. Now if a laser beam is shined into the back of the holographic film the object beam will be reproduced.

Unlike the pepper's ghost illusion this is considered a true hologram. It has features like motion parallax, which means the image changes as your perspective changes. It is also stereoscopic, where your eyes see two different images creating a perception of depth.

While this is an ideal method for our project there are many deterrents. For one this is a very high-cost method and procuring the necessary equipment is not an easy task, especially the laser and holographic film. Also, as a group this might be out of area of expertise. On the low end the cost would be a couple thousand.

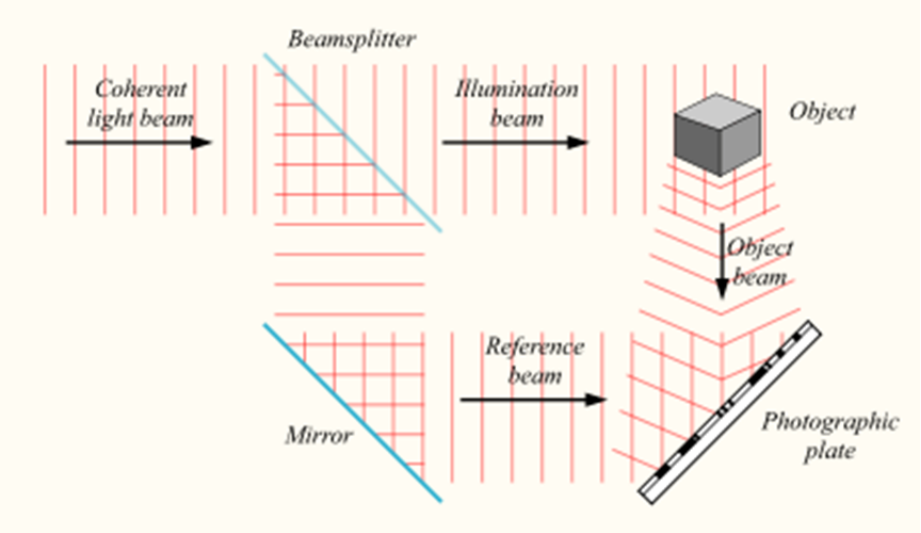


**Figure 19.** ***Recording a Hologram***

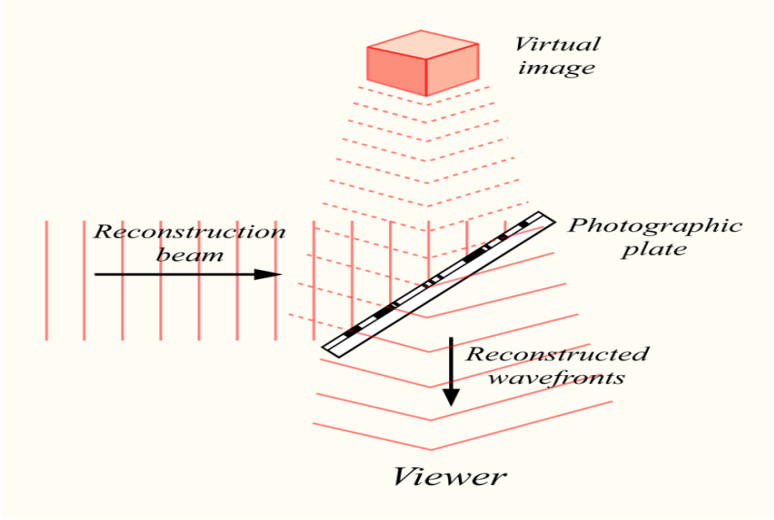


**Figure 20.** ***Reconstructing a Hologram***

### *3.2.1.7 Volumetric 3D Displays*

In comparison to the previous ways to create a hologram, a new method of displaying a 3D object drawn in a 3D space is through different types of volumetric 3D displays [4]. This technology is different from holograms, in that holograms are defined as an illusion of a 2D object in a 3D space. These types of displays were considered for the project, since we ideally wanted to have a 360 degree view of the HoloMon. However, since this technology is fairly new and under development, it was no longer under

consideration due to the limitations of our budget. Many of these displays can cost anywhere from $700 up to thousands.

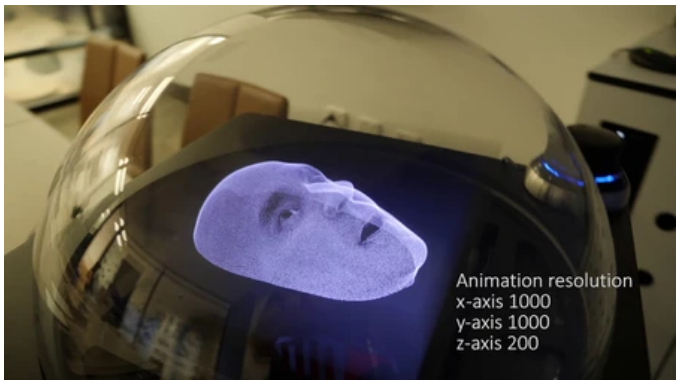


**Figure 21.** ***Volumetric display of a face***

#### *3.2.1.8 Holographic Display Technology Comparison*

| | Laser Hologram | Pepper's Ghost | LED-Hologram | Volumetric 3D Display |
|---|---|---|---|---|
| Motion Parallax | Yes | Yes | Yes | Yes |
| Stereoscopic | Yes | No | No | Yes |
| 360 Viewing Range | Yes | No, has some blocked edges | Yes | Yes |
| Cost | ~$1000s | ~$20 | ~$100 | ~$700-$1000s |

Table 3. Holographic Technology Comparison

The technology we decided to go forward with is the Pepper's Ghost implementation. The main deciding factor was the price. While it was not the best suited for the job, it would be sufficient to produce a suitable holographic effect for what is needed.

### *3.2.2 Microcontroller Unit*

There are many different systems applicable for the project as a microcontroller. Microcontroller Units (MCUs) are small computers that are on a semiconductor circuit chip. MCU's contain multiple CPUs with memory implementations such as RAM. The Microcontroller will be paired with some kind of computer based board.

### *3.2.2.1 Arduino Uno Rev 3*

Arduinos are small microcontrollers containing reset buttons, Analog pins for input, a 5V regulator, and many different pins. Arduinos are based on the ATmega328 [24]. In order to use this device, the user must install an IDE, pick out an Arduino, and then of course code. There is a USB port that is controlled by the USB-to-serial controller. The Arduino language is C/C++ which is a low-level coding language. The Rev 3 has a SRAM of 2kB and a program of 32KB.

Arduino Unos are a great component for users if they are new to using microcontrollers. The microcontroller has 14 digital analog inputs in which 6 can be used as PWM outputs and the other 6 can be used as analog inputs [24]. Once connected with a USB, the Arduino Uno can be used for almost anything and can easily access any computer. When evaluating the fragile state of the Uno, it can be noted that the Uno can go through most any kind of damage and survive. It is very durable and tough to damage if the user is taking proper care of the microcontroller [24].

Comparison of Arduino and the MSP430 is used to calculate which system best fits this project's needs. Arduinos are a great open-source that is easy to use both in the hardware/software aspect. There are different types of applicable inputs for the Arduino such as light, buttons, and messages. The following inputs can create an output such as a flashing LED or maybe like an app.

One thing to note is that this MCU, if selected, will be placed/arranged onto a PCB board that will be ordered when the building of the project starts. Certain aspects of the Arduino will be manipulated to fit the project's needs and certain parts of the Arduino may be removed if it's seen as unnecessary. Explanation for the future plans of Arduino will be stated later on.

### *3.2.2.2 MSP430FR6989*

The **MSP430FR6989** Kit is also a relatively easy to use system that evaluates a bunch of different modules. This is a simple platform that can perform coding, debugging, and measure any amount of energy. One example of how this software can be utilized is to count and display the numbers used through coding and the start timer. This microcontroller has nonvolatile memory of 128 kB with a RAM of 2kB [21]. There are a total of 83 pins and a 12-bit SAR [21]. The MSP430FRx has a lot more features compared to its brother MSP430Gx. The MSP430FRx has advanced sensing capabilities, LED functions, and a scan Interface. This is an astounding option for a microcontroller since it is capable of many functions and is easy to afford. This option is a little more expensive than the Arduino Uno by a couple dollars according to the Texas Instruments Website. This kit can be used to display shapes and or numbers by the user when coding. The MSP430FRx uses C and the following platform

that the user will code on is called Code Composer. After selecting the correct microcontroller options the user will be able to access programming capabilities.

### *3.2.2.3 MSP430G2553*

The **MSP430G2553** is obviously in the family of MSP430. It consists of many different features of peripherals with many different ways to be implemented. It is very quick due to its digitally controlled oscillator which allows the device to use a low-power mode. One thing to note about the MSP430G is that it has 16kB of non-volatile memory which is a decent size for this microcontroller. One useful feature that the MSP430FRx does not contain is the watchdog timer [20]. For the analog to digital converter there are 10 bits and about a total of ½ KB of RAM. To note, this RAM is essentially 4 times as small as the MSP430FR6989.

The MSP430G2553 is a very cheap option according to the Texas Instruments website, and ranges from 11.99 to 15 dollars with text [25]. However, even though this is cheap, due to the semiconductor shortage, it is almost impossible to find one of these without the delivery being delayed one year.

### *3.2.2.4 PIC Microcontroller*

The PIC Microcontroller stands for Peripheral Interface Controller and was developed in the late 1900's. This microcontroller was first formed to interface PDP computers to control a bunch of devices. PIC microcontrollers are a good piece of equipment of low cost and are a very fast executional program. Typical RAM for this device is 128 bytes. The following seen below is an Architecture Block Diagram of the PIC.

### *3.2.2.5 BBCMicro:bit V2*

One of the top rated microcontrollers is the BBC micro:bit V2 [45]. The BBC is very small and petite and has many unique features such as a microphone and speaker which make it efficient when dealing with audio-active projects [45]. Even though it is small it still contains LED status/matrix, speaker, touch sensitive logos, and buttons [45]. There is also a power saving mode and computing power component. Since the microcontroller does contain speakers, its functions are much more useful for any projects containing some sort of audio involvement.

| Microcontroller | Arduino Uno | MSP430FR6989 | MSP430G2553 |
|---|---|---|---|
| RAM | 2KB | 2KB | .5 KB |
| Storage/ Flash Memory | 32KB | 64KB | 16KB |
| Pins | 14 | 83 | 24 |
| Operating Voltage | 5V | 2V | 2V |
| Recommended Input Voltage | 7-12V | 1.8-3.6V | 1.6-3.6V |
| Clock Speed | 16MHz | 16MHz | 16 MHz |
| Width/Weight | 53.4mm/ 25g | 60 mm/ 40g | 60 mm/ 50g |
| Price | $44.08 | $47.74 | $11.99 |
| Functions | Controlling Relays, USB port, LEDs, and servos. | LCD, DMA, Scanning, Advanced Sensing, Scan Interface | Watchdog timer |

**Table 4.** ***Microcontroller Comparison #1***

| Microcontroller | Arduino | PIC | BBC Micro:bitV2 |
|---|---|---|---|
| RAM | 2KB | 4KB | 32Kb |
| Storage/ Flash Memory | 32KB | 14KB | 512 Kb |
| Pins | 14 | 14 | 20 |
| Operating Voltage | 5V | 2 - 5.5V | 1.8 - 3.6V |
| Recommended Input Voltage | 7-12V | 3.3 V | 3.3 V |
| Clock Speed | 16MHz | 64 MHz | 16 MHz |
| Width/Weight | 53.4mm/ 25g | Varies | 40 mm by 50 mm/ 25g |
| Price | $44.08 | $25.90 | $89.95 |
| Functions | Controlling Relays, USB port, LEDs, and servos. | Relay, buttons, LCD drive. | Built-in Speaker and Microphone, LED Functions, USB, Bluetooth |

**Table 5.** ***Microcontroller Comparison #2***

All of these MCU's are very compatible with almost any service. The BBC Micro:bit V2 and the Arduino Uno Rev 3 were the two final microcontrollers

debated in the review. The Micro:bit V2 has many different features and applications making it overall efficient for a variety of uses. However, due to the expense and the lack of simplicity the Arduino was the one selected. Another reason for selection upon this microcontroller is it has already been acquired thus making it the most affordable option for the project. The Arduino Uno Rev 3 is the overall most efficient microcontroller with its advanced features. This project does not require Bluetooth or speakers and in this case simplicity is better. Its applications are easy to use and the processor's core is the best out of the five.

### *3.2.3 Single Board Computers*

#### *3.2.3.1 Raspberry Pi*

Raspberry Pis are essentially small computers. There is an endless realm of possibilities with this hand sized model. With this computer, the user can create a gaming machine, build a server, control a robot, and basically any other kind of function.

Raspberry Pi is operated by a system called Linux which communicates between the computer’s hardware and its software [11]. The language used to develop websites and web applications is Python. An essential thing to note about the Raspberry Pi is its simplicity to the user whether the user is knowledgeable about Python or not. Raspberry Pi is a very good way to introduce technology while also learning without the need of a huge manual. There are hundreds of different experiments that can be done with this single board.

##### *3.2.3.1.1 Raspberry Pi Rev1*

The First revision of the Raspberry Pi was envisioned in 2006 and was inspired by the invention of the BBC Micro. The Raspberry Pi was not created until later 2012, then it became popular worldwide. The Raspberry Pi Rev 1 has a RAM of 0.51 GB and a 700MHx og CPU. The board is capable of ethernet, USB, and GPIO. This device used to be a lot more compact and bulky but has been adjusted to be smaller. It is still a lot bigger than the other Raspberry Pis. There are 4 USB’s on the device. This computer board is currently unavailable for purchasing and has a very long wait time on websites.

##### *3.2.3.1.2 Raspberry Pi Rev 2*

The revision of the Raspberry Pi to its Rev 2 made this device more compact and efficient. The Rev 2 has approximately 512 MB of RAM, an ethernet port, and two HDMI ports. The Rev 2 has an audio output jack that none of the other Revs contain. The GPIO has approximately 26 pins. The Rev 2 is also nearly impossible to obtain due to the semiconductor

shortage and due to the fact that the later Revisions are more efficient than the Rev 2.

#### *3.2.3.1.3 Raspberry Pi Rev 3*

The Raspberry Pi Rev 3 is the third evolution of the Pi. This revolution of the Raspberry Pi is a little smaller in size when comparing it to the Rev 1 [47]. It has a computer size of 1GB of memory and a RAM of 1GB. It has bluetooth capabilities as well as a USB, DisplayPort, Ethernet, and HDMI. One cool thing about the Rev 3 is that it has human interface input touch screen capabilities that the other Revolutions do not contain. When comparing it to Rev 1, the Rev 3 is approximately 10 times faster, which is significant. The price of the Rev 3 is approximately $200 and can vary depending on the website.

#### *3.2.3.1.4 Raspberry Pi Rev 4*

The Raspberry Pi Rev 4 is a little more clean looking and a tad more compact than the Rev 1. The Ram memory is about 2GB and the CPU model is Cortex. The Rev 4 was introduced most recently and has bluetooth, Wi-Fi, USB, Ethernet, HDMI, and GPIO capabilities. As noted, this Raspberry Pi has many more functions than the Rev 1 which is because it is the most recent renovation of the Raspberry 1. On many websites right now the Raspberry Pi Rev 4 is about $150-200 depending on the manufacturer. These small computers are very valuable, however, their prices have exponentially spiked due to the recent semiconductor shortage. This is the evolution of the Raspberry Pi that will be utilized for this project since the team has one already obtained. Luckily, it is also the best option for this project.

### *3.2.3.2 Orange Pi Zero*

This Orange Raspberry Pi is a good option for any users who are on a tight budget and or who need to implement semi-simple. It only has a memory of 512MB. This is only a half a gig of RAM. Which is really not much in comparison to the Raspberry Pi 4B which holds up to 8GB. The Orange Pi is a great platform for any android use. This single board computer also has video components as well as speaker features. The overall main reason to utilize this over 4B is that it is extremely cheap for those with a tight budget. The following seen below is a good representation of the devices communications Basis for the Orange Raspberry Pi.

### *3.2.3.3 Nvidia Jetson NANO*

The Nvidia Jetson NANO is a powerful compact computer that delivers a bunch of power to the module. This is a very efficient and high piece of technology. The NANO holds up to 4 GB of memory and has storage up to 16GB. These mini computers were built for AI involving computer vision

and media. One downside to using this product is that it is more expensive than usual due to its high functioning learning capabilities. Due to lack of supply in the demand chain, the prices for NANOs can range up to 400 dollars which is well above the financial plan for this project. Even though this computer is fast and has a wide variety of applications, it cannot be chosen due to its price.

The selected and specific Raspberry Pi has been shown in the following chart below. One main reason it was selected was due to the cost efficiency of the project. However, this main reason will not be applicable when working in the industry since money most likely won't be an issue.

| Raspberry Pi's | RAM | Functions | Price |
|---|---|---|---|
| Revision 4 | 2 GB | Ethernet, USB, bluetooth, Wi-Fi and GPIO | $150-200 |
| Revision 3 | 1 GB | Ethernet, USB, bluetooth, HDMI, DisplayPort and GPIO | $200 |
| Revision 2 | 512 MB | Ethernet, USB, HDMI, audio jack, and GPIO | NA |
| Revision 1 | .51 GB | Ethernet, USB, and GPIO | NA |

**Table 6.** ***Raspberry Pi Revision Comparison***

| Single Board Computer | Raspberry Pi 4B | Orange Pi Zero | Nvidia Jetson NANO |
|---|---|---|---|
| GPU | Nvidia GeFOrce GTX 1080 | Multi-core GPU Mali T720 | 128 NVIDIA |
| CPU | ARM11 | H6 64 bit 1.8 GHZ Cortex- A53 | Quad-core ARM Cortex-A57 |
| Recommended current (at Idle) | 575 mA | 90 mA | 2 A |
| Memory | 8 GB | 512 MB | 4 GB |
| Card | Micro SD card | Micro SD card | Micro SD card |
| Pins | 11 | 26 | 40 |
| Price | $188.76 | $ 62.99 | $ 399.00 |
| Functions | Multimedia performance, easy connectivity, fast processor. | Simple/easy User accessibility, android function, servers. Speakers, videos. | Robotic functions, low power features, recorders, IoT applications |

**Table 7. *Computer Comparison***

As seen above, these three single board computers all have many different features making them useful for this project in particular. Since the Raspberry Pi is more user friendly to beginners, this computer was chosen. Another reason this computer was chosen was due to the fact of the supply chain issues and demand of the product causing very competitive and high prices for all of the computers. Since the Raspberry 4B has already been obtained by one of the team members, this is an easy/affordable choice for this project.

### *3.2.4 Raspberry Pi and Arduino Connection*

Now having two of the most important hardware selected the evaluation of the connections are important. The Arduino is a microcontroller while the raspberry pi is a microprocessor, both very important in this system. How these two boards connect is by the USB connector on the Raspberry. Once using the USB port, it will be easy to upload any code from your computer (Raspberry Pi) and transmit it to your Arduino.

Overall looking at the bigger picture of why should these two be together? Why not just use a Raspberry Pi? One reason to use the Arduino as well is because it can convert analog-to-digital with its ADC chip. There will be specific software that will need to be used in order to do this. The arduino will essentially act like a server node while the Raspberry Pi takes data from the nodes and sends the Hologram using the USB. A breadboard will be used with wires connecting the whole system. There are a couple different softwares applicable to this section of the project, however, it will not be discussed until later.

### *3.2.5 Serial Communication Protocols*

For communication between our microcontroller and our ancillary units we need some form of serial communication to transmit data between them. The 3 most common serial communication protocols that are supported by many devices are UART, I2C, and SPI.

#### *3.2.5.1 UART*

UART is a hardware communication protocol utilizing asynchronous serial communication. Serial communication means that one bit is transferred at a time and asynchronous means that there is no clock signal to synchronize the bits from transmitter to receiver, this implies that bits can be lost under certain circumstances.

The baud rate in most cases is set to 9600. Baud rate is the rate at which information is transferred through a communication channel and is in the units of bits per second. As a result of being asynchronous, the baud rate of each device needs to be set to the same to main data integrity.

A frame is the unit of transmission in UART, and it is usually 8-bits. Framing helps the receiver identify units of data. The frame consists of a

start bit, data, optional parity bit, and stop bits. The start bit is held at high voltage when there is no data transmission, and it is pulled to low when transmission starts. The parity bit tells us if the data is even or odd and checks if it has changed during transmission. At the end of the frame is the stop bit which pulls the voltage back from low to high.

The most popular UART configuration is:

- 9600 baud
- 8-bit data
- No parity bit
- One stop bit
- No flow controls
- LSB first.

### *3.2.5.2 I2C*

I2C uses a bus topology and is a two-wire serial interface. It was introduced by Philips (now NXP) and became popular after its patent expired in 2006. I2C uses two connection lines called serial data line (SDA) and serial clock line (SCL). It follows a master-slave architecture. The master always initiates communication first and drives the clock. The bus uses half-duplex (bidirectional lines), meaning the communication can happen in both directions, but only one way at a time. Typically, there is only one master and multiple slaves, however, multiple masters are allowed. Since all the devices are attached to the bus, they each need to have their unique address.

The lines are pulled to high by default. Transmission occurs by pulling low and releasing high. There are several I2C modes which affect the transmission speeds.

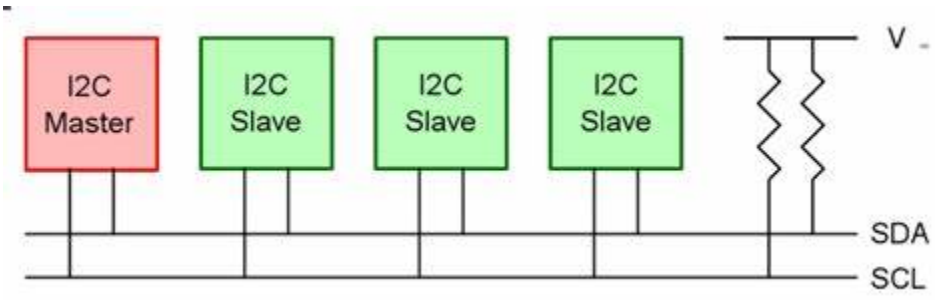


**Figure 22.** ***I2C Diagram***

| I2C modes | Transmission Speed (up to) |
|---|---|
| Standard Mode | 100 kHz |
| Fast Mode | 400 kHz |
| Fast Mode Plus | 1 MHz |
| High Speed Mode | 3.4 MHz |
| Ultra-Fast Mode | 5 MHz |

**Table 8.** ***I2C Modes***

I2C data packets are transmitted in 8-bit sections and encapsulate the slave address, register number, and data to be transferred. I2C follows specific read and write protocols which include start and stop conditions and acknowledge/not acknowledge bits (ACK/NACK).

When the master initiates transmission the start protocol allows the SDA line to shift to toggle from high to low state while SCL is high. The stop condition is also sent by the master to signify the end of the transmission. The protocol toggles the SDA line from low to high while SCL is high. A repeated start condition can happen when another start is sent without needing the stop condition. This happens to prevent another master from taking control of the bus.

After the Master sends the start condition it sends a byte which contains the address. More specifically, the first 7 bits correspond to the address and the last bit signifies whether it is a write or read call. Due to the bus architecture, all the slaves listen to the first 7 bits of the address byte and only the device that matches with it will read the last bit. The device will assume all data is meant for it until the stop bit is asserted.

### *3.2.5.3 SPI*

Serial peripheral interface (SPI) is a synchronous, full duplex protocol. It was introduced by Motorola (now Freescale). It is full duplex, meaning that it can transmit in both directions simultaneously from master to device and vice versa. It is synchronous meaning that the data is transmitted on the rising or falling clock edge. SPI can either be a 3-wire or 4-wire interface, the 4-wire interface is more popular.

The 4-wire interface contains 4 signals:

- Clock
- Chip select
- MOSI
- MISO

The clock signal is generated by the master node and transmitting the data to the sub nodes is synchronized with the clock. Compared to I2C, SPI can support higher clock frequencies. The chip select signal from the master node is used to select the sub node. The signal is pulled to high to disable the sub node device. MOSI is the data line that transmits data from the master to the device. MISO is the line that transmits data from the device to the master.

For our use we will most likely have a multi-subnode configuration. Multi-subnode configurations can be set up using only one master and can be connected regularly or daisy chained. In the regular mode there is a chip select for every subnode. If multiple chip selects are enabled, then the MOSI line can have corrupted data. Therefore, as more devices get added this configuration becomes less desirable. In daisy chaining mode, there is only one chip select that is tied to all the devices. In this configuration data is shifted through each subnode until it gets to the position that it needs to. Daisy-chaining may or may not be supported by systems that support SPI, so referring to the documentation is necessary.

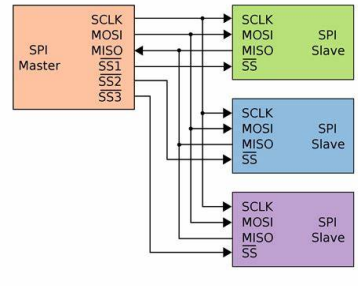


**Figure 23. SPI *Diagram***

| UART | SPI | I2C |
|---|---|---|
| Asynchronous | Synchronous | Synchronous |
| Simplex, Full Duplex, or Half Duplex | Full duplex | Half Duplex |
| Baud rate (bps):1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 | Can go over 10Mb/s | 3.4 Mb/s (High Speed) |
| 2 data lines (RX, TX) | 4 data lines (CLK,MOSI,MISO,CS) | 2 data lines (SDA and SCL) |
| Data lines bidirectional | Data lines bidirectional | Data lines bidirectional |
| Slowest, easiest to implement | Faster, but harder when more than one sensor | Fast and simple to implement |

**Table 9.** ***Serial Communication Comparisons***

### *3.2.6 Gesture Recognition*

Another big portion of our project is the use of gesture-based recognition so that a user can interact with the digital HoloMon. Below are separate areas of research to see what are the best hardware components and protocols to use for our project.

For the purposes of understanding hand gestures, they can be classified into two categories: static and dynamic. Static, as its name implies, refers to the stable hand position / posture of a hand. As stated in [2], dynamic classification refers to “a series of hand movements within a gesture,” such as waving.

#### *3.2.6.1 Wearable glove-based sensors*

The first of the two technologies is wearable glove-based sensors. These gloves were suited up with various sensors that would collect hand motion

and position via detecting the correct coordinates of the location of the palm and fingers. The glove was then connected to a computer interface with a cable. This technology usually involved flex sensors that could detect flexing of the hand, an accelerometer to track how fast the hand was moving, and a gyroscope that figures out the xyz movement of the glove. Despite the gloves providing quality results, researchers quickly realized that it would be unsuitable for certain age groups. The elderly might find discomfort with using gloves that they could not take off in addition to the wired connection that makes this technology unnatural. People with sensitive skin or who have suffered burns would also be unlikely to use gloves to track hand gestures. Children within certain age groups might also find the gloves to be cumbersome and clunky.

#### *3.2.6.2 Camera vision-based sensors*

The glaring issues with the first technology led to the development of computer vision techniques using cameras. Many algorithms have been programmed in attempts to find the most suitable way to detect hand gestures with computer vision. The following seven are the most reported approaches to the problem such as color, appearance, motion, skeleton, depth, 3D-model, and deep-learning [2]. There are a lot of factors involved in the efficacy of the aforementioned algorithms which include but are not limited to: type of camera used, resolution of the processed image or video, type of segmentation technique, recognition rate, number of gestures, and detection range. For this method, different types of camera configurations can be used such as monocular, time-of-flight (TOF), fisheye, and infrared (IR).

Some challenges that occur while using camera vision-based sensors will influence the design constraints if this project considers using cameras for the gesture recognition portion. There might be issues with lighting variation, multiple people in the background, effect of occlusions, too complex of a background, other objects appearing as hands, and trade-offs that happen between processing time against resolution, frame rate, and foreground/background objects.

Before choosing a type of gesture recognition to go with, each of the following algorithms will be discussed in detail. This is to ensure that if we choose to implement a camera computer-vision based approach, that we can pick the most suitable algorithm to pair with it for our needs. There are two ways to write a color-based gesture recognition. The first is by using a glove marker, where the camera tracks the movement of the hand by using a glove with different color marks. The detection of these particular colors allows the camera sensor to track the location of the palm and fingers, outlining the geometric model of the shape. While this method is better than using the first type of wearable sensor since it does not require to be tethered, it still requires the user to wear colored gloves. The other method of color-based recognition is by skin color, which is one of the

most popular methods for hand segmentation. This is achieved via pixel based skin detection by comparing each pixel in an image to be classified into skin or not individually from its neighbor. Another way of achieving this is by looking at the skin pixels and spatially processing them based on information such as intensity and texture.

The next algorithm is appearance-based recognition. This method relies on extracting the image features to create a machine learning model that anticipates the visual appearance of a hand and comparing these parameters from the input image frames. This algorithm is particularly useful because it can detect various skin tones and solve the occlusion issue by separating into two models, namely the motion model and a 2D static model.

Another algorithm under consideration is motion-based recognition. This method primarily relies on the AdaBoost algorithm by extracting the object through a series of image frames. The main issue with basing gesture recognition off of motion is that it is particularly susceptible to misinterpreting one gesture as one or more gestures. If there are moving objects in the background as well, it can misinterpret those as gestures too.

Skeleton based recognition seems to be the most promising for our project. This algorithm focuses mainly on the geometric and statistical features, such as the “joint orientation, the space between joints, the skeletal joint location and degree of angle between joints and trajectories and curvature of the joints” [2]. One of the downsides for skeleton based recognition is the range of detection for certain cameras that are not specialized in tracking joints.

Depth-based recognition is based on using different types of cameras, which may not be suitable for our project given our limited budget. This approach uses a depth camera that can provide 3D geometric information about an object. Similarly, the 3D model-based recognition approach also utilizes some volumetric depth. This is different from depth-based recognition though, since 3D model-based recognition compares the input image with the 2D appearance projected by a 3D hand model. Another downside to using these algorithms is that 3D models require a large dataset of images to formulate virtual shapes of a hand, and the matching process comparing 2D and 3D consumes a lot of time and computation power. In this last facet, deep-learning based recognition is similar in requiring a large dataset for the model to train off of. However, artificial intelligence provides a reliable method in outputting a good prediction.

Based on this information in [2], the following cameras will be under consideration for part selection: Logitech HD Pro Webcam C920, Kinect, mobile device camera.

| Gesture Recognition Hardware Selection | | |
|---|---|---|
| **Criteria** | **Wearable-Based** | **Camera-Based** |
| Cost | $70+ | $20+ |
| Convenience | Uncomfortable for user to wear, tethered by wires | Natural for user to use since it does not require them to wear anything |
| Accessibility | Not very accessible due to the fact that there is no “mainstream” product that can be rented or bought | Very accessible since webcams and cameras are prolific for commercial/private use |

**Table 10. Gesture Recognition Hardware Comparison**

## *3.2.7 Cameras*

As previously aforementioned in the last section, the following cameras that this subsection seeks to expand upon are the Logitech HD Pro Webcam C920, Kinect, and mobile device cameras.

### *3.2.7.1 Logitech HD Pro Webcam C920*

The Logitech C920 HD Pro Webcam has a special feature that allows it to do well in low light. Its video capture resolution is 1080p. It connects to any host device with a common USB-A connectivity technology. It conducts image stabilization and zoom type digitally. Its video capture format is in audio video interleave (AVI) which allows it to store both video and audio data in a single file. The Logitech C920 HD Pro’s image capture speed is 30 frames per second (fps). It also includes proprietary technology called Logitech Fluid Crystal™ that “ensures that each video call or recording offers crystal-clear images with smooth, fluid motion and rich, true-to-life colors, even in real world conditions,” [26].

This webcam is primarily under review since it is a camera that the team currently has in possession, making it the most affordable choice to pick. It also meets all the criteria that the team is looking to use for camera-based gesture recognition techniques, making it overall a strong candidate to choose from.

#### *3.2.7.2 Kinect*

Kinect is a well-established brand of motion sensing input devices produced by Microsoft for the Xbox, similar to Nintendo's Wii Remote and Sony's PlayStation Move. The devices generally contain Red-Green-Blue (RGB) cameras with infrared projectors and detectors that have the capability to map depth through either structured light or TOF calculations [27]. It is a strong candidate used to perform real-time gesture recognition and body skeletal detection approaches. Additionally, like the Logitech C920 HD Pro webcam, it contains microphones. The Kinect v1 has a camera with RGB 640 x 480 pixels at a 30 fps refresh rate with a depth camera of 320 x 240 pixels. The Kinect v2 has a 1920 x 1080 pixels at 30 fps with a depth camera of 512 x 424 pixels. Another major difference between the two is that the Kinect v2 greatly improves on the field of vision (FOV).

Both the Kinect v1 and Kinect v2 for Xbox 360/One, respectively, have been discontinued, the only sensors left are renewed and higher priced than in previous years. This would make it difficult to acquire, since Microsoft does not make any models anymore. The running price for a Kinect Sensor for the Xbox 360 is $67. There is, however, a newer model of Kinect called the Azure Kinect. This latest Kinect was released in 2019, seeing its continued success as a quality motion sensor in non-gaming applications such as academia and other fields. The technology in the Kinect has since spread out to other areas at Microsoft such as being integrated into the Hololens. The point of the Azure Kinect is to offload some of the computational work from the device and allow for more powerful features through Microsoft's Azure cloud computing service such as using artificial intelligence [27]. This integration would improve the accuracy of the depth-sensing and reduce the power demand which would lead to more compact units. However, since our project does not use Azure, and with the price point of it being $400, this would make the Kinect an overall unlikely part to be selected.

#### *3.2.7.3 Mobile Device Cameras*

This section is dedicated to the camera phone on a mobile device. Mobile computing has evolved significantly since the early 2000s. Since there are so many versions of camera phones, this section will detail them in broad terms. While quick to record and have amazing resolution, it would take a little more work to utilize them solely as a source to capture and send data in real time to the single-board computer to which the camera is attached. Additionally, the FOV on a given mobile device would severely limit the area in which a user can conduct gestures. Despite camera phone hardware and software being top-tier in the modern day, having the camera app for long periods of time would drain the power supply quickly and that detriment is a significant setback to choosing mobile device cameras. Additionally, it would be unreasonable to purchase a $400+

phone to only be used as a camera. The camera phone used for reference in the below table

| Camera | Resolution | Frame Rate | Price |
|---|---|---|---|
| Kinect v1 | 640 x 480 | 30 fps | $70 |
| Kinect v2 | 1920 x 1080 | 30 fps | $70 |
| Azure Kinect | 3840 x 2160 | 30 fps | $400 |
| Logitech HD Pro Webcam C920 | 1920 x 1080 | 30 fps | $0 (acquired) |
| Camera Phone | 3840 x 2160 | 60 fps | $700+ |

**Table 11. *Comparison of Cameras***

### *3.2.8 Housing Unit*

The Holomon system will be positioned stationary on a flat surface and will need to be self-contained. The housing unit will encapsulate the projected hologram and the chosen display that would be used to create the image. Design and construction of the housing unit will vary depending on the method we use to create the hologram. This section will assume we are using the Pepper's Ghost method.

In reference to the Pepper's ghost method in section 3.2.7, a pyramidal shape made out of transparent material is needed to create the effect of a hologram. The main purpose of this section is to explore what is the optimal material to use. It is important to note that higher transparency will result in better hologram quality. Transparency depends on the linear passage of light through an object. An object's transparency can be measured by its total transmittance. Transmittance = I/Io, where I = transmitted light and Io = incident light.

| Material | Light Transmittance |
|---|---|
| polycarbonate (PC) | ~88% |
| (polymethylmethacrylate)PMMA | ~93% |
| Polyethylene terephthalate (PET) | ~88% |
| Polyvinyl chloride(PVC) | ~82% |

**Table 12. *Comparison of light transmittance of materials***

PMMA had the highest light transmittance with 93%. PC and PET were roughly tied with a transmittance of 88%. PVC had a transmittance of 82% which was the lowest out of the considered materials.

### *3.2.9 Display*

Summarized research shows that there is no standard way to create a hologram, with private companies creating their own displays with customizable resolution and extraneous technical specifications. A High Definition Multimedia Interface (HDMI) connection is preferable, since this technology carries signals at 144Hz at 1080p for high resolution and has the option of adding audio to the HoloMon if desired. Digital Visual Interface (DVI) is another option, but many newer devices and models do not carry this type of video input connector. A DisplayPort (DP) would provide the best connection for both audio and video signals since it can transmit 144Hz up to 4K, however most assets currently work best in 1080p so an HDMI connection will be sufficient.

Given this information, a TV with HDMI connection or a PC monitor with HDMI connection would be the best possible choices. Other choices to consider are tablets and smartphones. The dimensions of HoloMon's design will be the biggest influencing factor of what display will ultimately be best.

| Display Unit | | | | |
|---|---|---|---|---|
| **Criteria** | **TV** | **PC Monitor** | **Tablet** | **Smartphone** |
| Cost | $100+ | $100+ | $300+ | $500+ |
| Resolution | 1920 x 1080 | 1920 x 1080 | 1920 x 1080 | 1080 x 2340 |
| Connection | HDMI | HDMI | USB-A | USB-C |
| Relative Size | Extra Large | Large | Medium | Small |

**Table 13. Display Unit Comparison Table**

### *3.2.10 Power Sources*

There are many components that will need power source selection for the HoloMon: TV monitor, Raspberry Pi, and the Arduino. When selecting the correct power source to use for electronic components it is important to acknowledge the voltage/current input and output. Raspberry Pi 4B usually outputs 540 milliamps and consumes 2.7 Watts at idle state. The chosen Raspberry pi can delegate 3 amps, but has a minimum current of 2.5A. There is also the display, either a TV or monitor, that will require a power supply as well. However, it is probably best to also use the wall outlet for the display since it is stationary and requires a lot of voltage to output a moving image for long periods of time.

#### *3.2.9.1 Wall Outlet (For Raspberry Pi or TV Monitor)*

The Wall Outlet will be in consideration for the Raspberry Pi and or the TV Monitor. Since this project does not require any movement, the Raspberry Pi will be plugged into the wall (delegates 15-60 A). There is a special plug head that ensures that the right amount of current is flowing Raspberry Pi. If too much current went into the Raspberry Pi, the Port could be ruined and or the whole device could be fried. There is a resistor on the device that can down convert the input current if it is too high, however just for safety we utilize the plug head as well (see below for voltage regulations).

### *3.2.9.2 LABISTS(For Raspberry Pi)*

The LABISTS has an input voltage of 100-240V and an input frequency 50/60 Hz according to the label inside of the plughead. The power supply has a 0.5 A max of input current and can supply 300 mA to the Raspberry Pi. The switching power supply can give 5V towards the Raspberry Pi. The LABISTS has also been acquired making it the most affordable option compared to all the other plug heads.

Another thing to note on the power side of the Raspberry Pi is how to properly shut down the device. It is not safe to simply unplug the Raspberry Pi. The Raspberry Pi must be shut down using a sudo command on python. Another possibility is on the plug head there is a button that has the option to turn the Raspberry Pi on and off.

### *3.2.9.3 Canakit (For Raspberry Pi)*

Another viable option for the power supply for the Raspberry Pi power supplier is the Canakit. Canakit is a cheap option and in line for the budget. This power supply has a minimum voltage of 100V and can supply 30 Watts. The item is also small and compact which is good in order to keep the project's weight to a minimum. The Cankit is a 5 foot cable with a 5 V DC/ 2.5 A regulation.

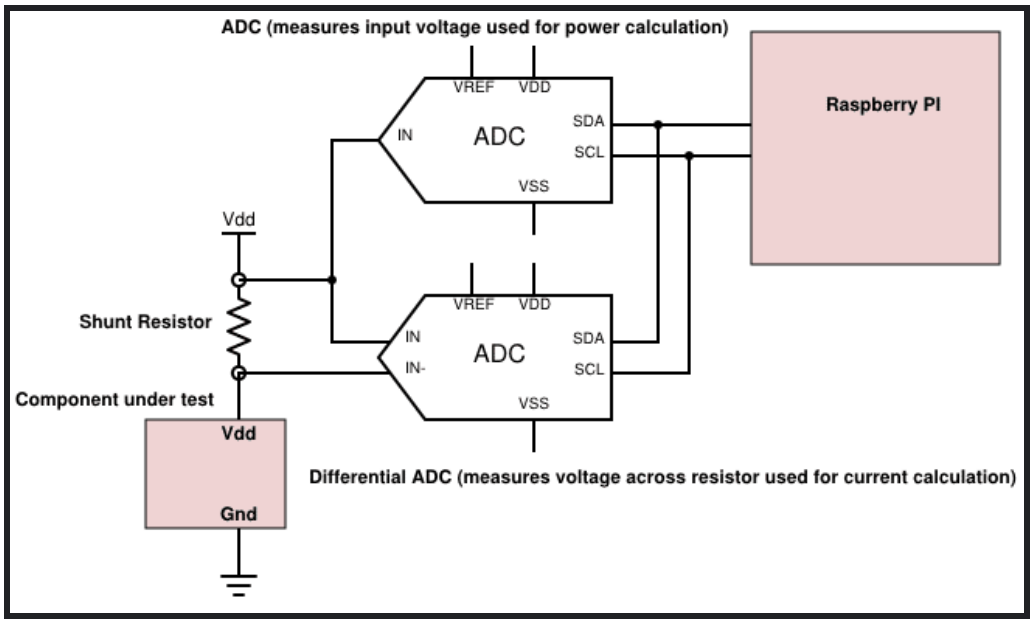


**Figure 24.** ***Raspberry Pi Voltage Regulator***

### *3.2.9.4 Li-ion Battery Pi HAT(For Raspberry Pi)*

In comparison to the power source being an outlet, the project could also utilize Battery Pi Hat. Battery Pi Hats are typically utilized in projects with a very mobile function such as a drone and or any moving creations. However, since the Hologram is stationary, there will be no such need for this battery source. The Li-ion battery that ouputs 5V and it uses a 14500 battery. An interesting thing about the HAT is that it can also charge the battery with indicators on its battery life using LED's.

### *3.2.9.5 9 Volt Battery (For Ardunio/PCB Board)*

One thing to note is that the Arduino must also have a power supply. For the project the Arduino will be powered by the Raspberry Pi which is powered by the wall. However, the PCB will still contain a power jack just in case for independent application and testing. For every electronic device there is a voltage source. The Arduino will be placed on a PCB board and must have some kind of power source. The easiest way to supply power is to set up a power jack to the EAGLE schematic and to give the PCB power that way. There are other applications of connecting the batteries to the PCB board, but the easiest way to calculate the PCB is the power jack since it is more simple to add onto the schematic and if something goes wrong with the final design portion, the team will not have to order a completely new PCB board. 9V batteries are obviously very high in voltage and a lot higher than what the Arduino Rev 3 would use, however, there will be a voltage regulator attached to the PCB board that will allow the battery source to only give 5V to the board. This voltage

regulator will be explained in Chapter 6, overall, the voltage regulator will use a feed-foward design and possibly include negative feedback. Voltage regulators can be used in AC or DC voltages, since we are using a battery(DC) this will be able to regulate our voltage perfectly. 9 Volt batteries are very cheap and can be obtained in a pack of 8 for about $12.

| Power Source | LABISTS | Cankit | Li-ion HAT |
|---|---|---|---|
| Input Voltage | 100-240V | 100V | 168 V |
| Output Voltage | 5V | 5V | 1.2V |
| Input Current | 2.5A | 2.5A | 2.4A |
| Connection Type | Micro-USB | Micro-USB | On-board connection |
| Price | $11.98 | $9.95 | $23.99 |

**Table 14.** ***Comparison of Power Sources for Raspberry Pi***

Since there are three different electronics that need to be powered, there will be different power sources for the project. First, the Raspberry Pi and the TV monitor can be easily plugged into a wall outlet and operated this way. For the Raspberry Pi, the best option in this case for the power source is the LABISTS power head. Overall, the three power sources compared for the Raspberry Pi are nearly identical besides the Li-ion HAT. The Li-ion HAT would be perfect for any mobile projects, but due to the simplicity of the project it is best to use a plug-head cord connecting to an outlet. Another reason the LABISTS has been chosen as the power source is because it has already been acquired making it the cheapest option.

Finally, for the Arduino, the power source selected is simply a 9V battery. In the PCB prototype it was decided that a power jack will connect the 9V battery to the rest of the PCB. This is the simplest option and makes the PCB easier and safer to operate so that parts get ruined and have to be replaced. One constraint for this is that the battery will have to be replaced eventually by the user.

| Electronic Device | Power Source |
|---|---|
| Raspberry Pi | Wall Outlet/ LABSIST plug head |
| TV Monitor | Wall Outlet |
| Arduino/PCB | 9V battery (independently) or Raspberry Pi (Project) |

**Table 15.** *Final Selected Power Sources for Raspberry Pi/ TV Monitor/ Arduino*

### *3.2.11 Cloud Services*

HoloMon is connected to the user by both a mobile app and the hardware projecting the hologram, the bridge between both of those connections would be an API connected to a database. There are two common ways to serve an API: through a dedicated server or a cloud provider. Using a cloud server is beneficial due to its low cost, scalability, and abundant resources. Given our requirements, there is no need to rent rack space.

The three main cloud providers are Microsoft Azure, Amazon Web Services (AWS), Google Cloud. The initial iteration of HoloMon would not require more than 2GB of storage and less than 1GB of RAM to run without any delays, this would open the door to the free tier virtual environments given by any of the cloud providers. The comparison of the three is stated below.

| AWS | Azure | Google Cloud |
|---|---|---|
| 10GB Storage | 500GB Storage | 10GB Storage |
| 10,000 Git requests monthly | 5G Outbound data | 50GB outbound data |
| 750hr/month (EC2) | 400 requests to storage/month | 2m HTTP invocations |
| 1GB RAM | 1M requests/month | 1TB of queries |

**Table 16.** *Cloud Provider Comparison*

Community is also something to consider when choosing a cloud provider, as resources such as documentation, tutorials, and other developers are extremely important when dealing with new technology. These are the results of Stack Overflow's 2021 survey on professional use of cloud technologies.

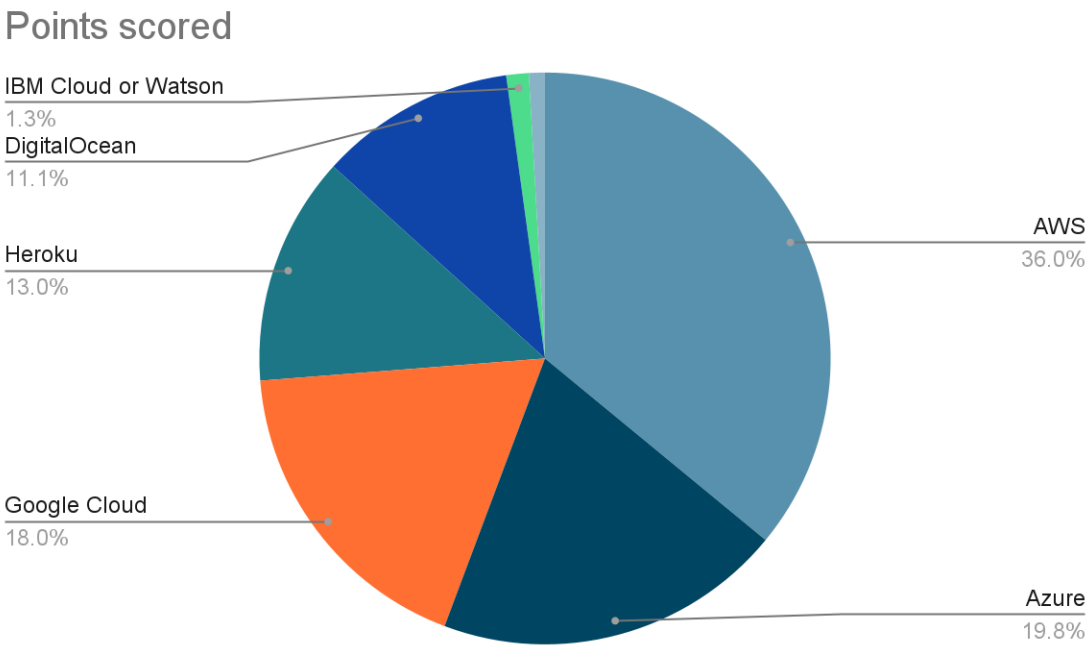


**Figure 25**. *Stack Overflow 2021 Survey Results*

Both the API and Database server would be running within a cloud provider virtual instance where we can SSH to modify. The instance can then be open to any HTTP calls requested by the mobile client and the hardware.

### *3.2.12 Backend API*

The whole reason behind the API is to separate the business logic and any authentication or sensitive information from the client. The main requirement would be a framework to accept HTTP requests and return responses as the user will be able to change their profile information and HoloMon assets through such protocol. There are a variety of back-end frameworks to potentially choose from and most would do the job well given our requirements. The main requirements for the framework would be the compatibility with the relational or non-relational database we would be using to store the user's information as well as the compatibility with the cloud instance and its ease of use.

Just as with the cloud provider, the biggest difference between the frameworks would be the community behind its development. The community size usually indicates the number of resources available to master the technology as well as its maintenance. Below is the popularity of the programming, script, and markup languages entered in the Stack Overflow 2021 survey [3].

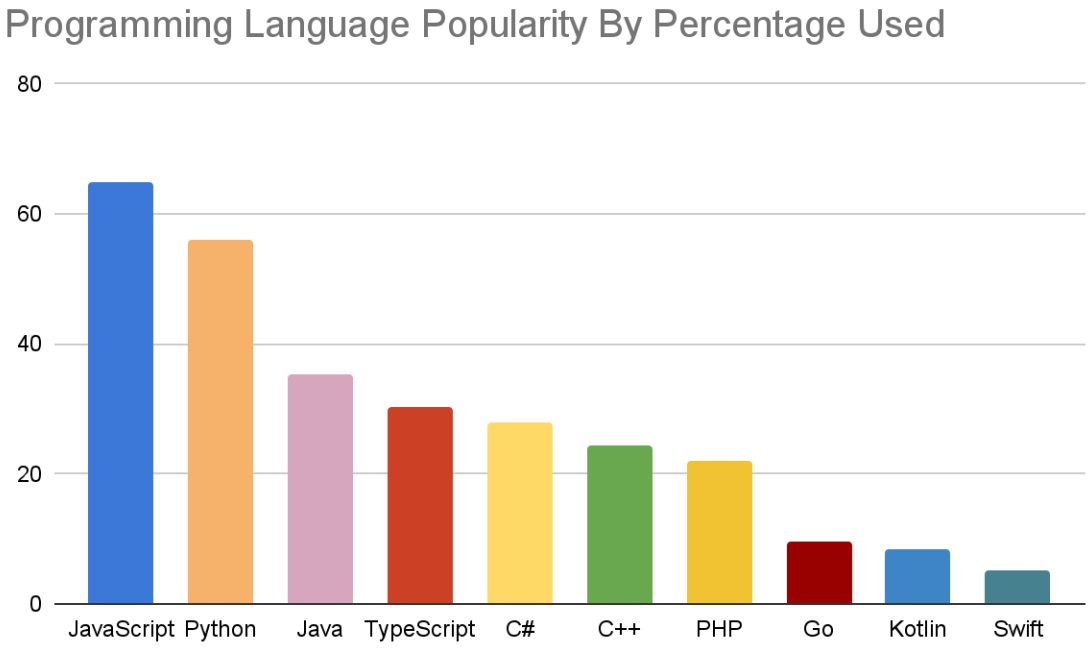


**Figure 26**. *Stack Overflow 2021 Survey Results pt.2*

By narrowing it down to the top three, JavaScript, Python, and Java would be the most popular programming languages used. All three are capable of fulfilling our server requirements and have frameworks to aid the process.

#### *3.2.11.1 JavaScript*

According to the survey, JavaScript is the most popular language used in a professional environment. The most popular JavaScript server-side backend framework would be Express.js

Express currently holds the most GitHub stars of all other JavaScript backend frameworks. It lets users create APIs and connect to databases by using Node.js as a JavaScript runtime environment. It is mainly used to develop Web Applications and REST APIs in the most minimal way possible and with the least amount of boilerplate code. FastAPI is also extremely easy to pick up and to create functionality as you develop, due to its less opinionated nature.  According to the State of JS, it has been the most used JavaScript framework since 2017.

#### *3.2.11.2 Python*

Python is widely known for its ease of use and an excellent multi purpose programming language. Python would be a logical choice since the machine learning model is already written in the language, this would take away a lot of complexity from having to learn another programming language.

Many python backend frameworks could achieve our requirements, the latest being FastAPI. FastAPI is a high-performance web framework for building APIs with support for Python 3.6 and later. Not only does it require very minimum lines of code to fully develop a REST API, but it also enforces python’s type system and eliminates a large number of errors that one would encounter by using an interpreted language.

#### *3.2.11.3 Java*

Java has been around for a serious amount of time and has matured tremendously. Though it is not the most modern of the three, it has the edge of being the primary programming language of universities, therefore, everyone involved in the project would be familiar with its syntax. Java can be used as a backend server-side language with the combination of the Spring framework. Spring is Java’s most popular framework, with an emphasis on speed, ease of use, and security.

Spring allows for minimum boilerplate code while maintaining its intuitive and readable methods. Spring also supports easy methods of connecting to SQL database servers, filling up all the requirements necessary for the project.

### *3.2.13 Database*

There are two main types of databases commonly used with REST APIs, SQL and no-SQL databases.

#### *3.2.12.1 SQL Databases*

SQL databases are by nature, relational. They are often used to describe entities by their relations and connect them with joint tables. They are also predetermined by a schema, meaning that the data structures have to be designed and pre established. SQL databases scale vertically, are table-based, and are better for multi-row transactions [6]. Commonly used SQL databases are MySQL, PostgreSQL, and SQLite.

SQLite is fully open-source, known for portability, reliability, and strong performance in low-memory environments. SQLite's biggest advantage is how user-friendly and portable it is, while its biggest disadvantage would be its limited concurrency [6].

MySQL is the most popular database management system since 2012 [6]. Just as SQLite, it is designed for both speed and reliability and powers gigantic web applications such as Twitter, Facebook, Netflix, and Spotify [6]. MySQL's main advantages would be its popularity, security, speed, and ease of replication. The disadvantage would be the slowed development.

PostgreSQL, or Postgres, is currently the most advanced open-source relational database. It follows its goal of being highly extensible and standards compliant. Unlike the previous two, Postgres could also work as an object-relational database, giving it extra flexibility for both types of database systems. Some of Postgres' advantages would be its tight SQL compliance, the open-source nature (completely free for production use), and its extensibility by using dynamic loading. Some of the disadvantages would be the lack of memory performance and its lack of popularity compared to MySQL [6].

#### *3.2.12.2 NoSQL*

NoSQL databases, by their name, are databases that do not use the SQL structure of storing data. These databases use a non-relational object structure and have dynamic schemas with unstructured data. NoSQL databases are horizontally scalable and based on document, key-value, graph, or wide-column data structures [6]. Due to its lack of structure, insertions and deletions are often faster with NoSQL databases. These databases are often used by large tech companies to store their gigantic, unordered, datasets [9].

The most commonly used NoSQL database would be MongoDB. MongoDB uses a document-oriented data model while using an unstructured query language [10]. The biggest advantages of MongoDB

would be its high speed, simplicity, easy learning environment, documentation, and scalability. Its biggest disadvantages would be its failure to support joining data, making the task tedious and performance lacking, and its high memory usage [10].

In both cases, developer popularity is highly desirable due to the number of documentation and guides available to learn and develop with the technology. The graph below displays the popularity of the stated database management systems by the percentage of professional use.

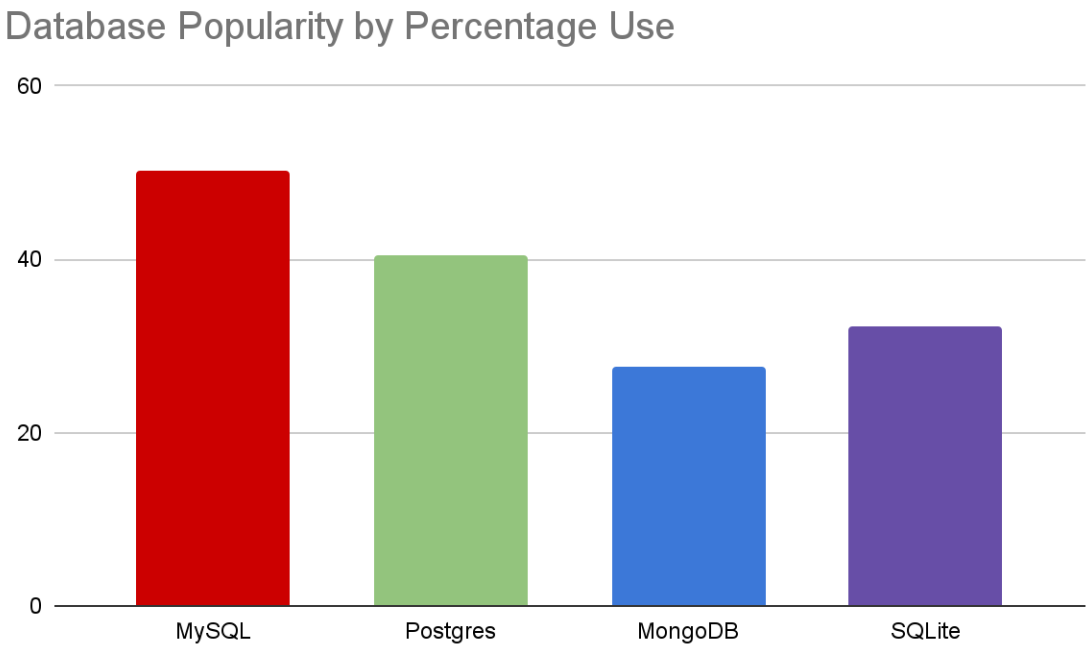


**Figure 27.** *Stack Overflow 2021 Survey Results pt.3 [3]*

## *3.2.14 Client front end mobile app*

The client mobile application provides the user with an interface to alter their information as well as the HoloMon assets to be displayed by the hardware. The main purpose of this portion of the project is to connect to the cloud provider instance or dedicated server through the HTTP protocol. There are two main approaches to developing mobile applications: Cross-platform or native.

### *3.2.13.1 Native Development*

A mobile application is exclusively built for a single platform when built natively. The two main platforms are currently Apple's IOS and Google's Android, both of which have different app stores, APIs to connect to the hardware, and programming languages supported.

The main advantage of native mobile development is the exceptional user experience that comes with its high performance. Native applications also

visually conform to the main platform's UI, giving the user an application that feels professional and in line with their phone's UI [12].

The main drawback would be the cost and time of development. To achieve the most complete user reach, two code bases would have to be maintained, with two completely different design patterns, standards, development environment, and programming languages. Native development is usually the method used by large companies with multiple development teams.

IOS development is dominantly developed in the Swift programming language created by Apple. Previously IOS developers used objective-C to create applications and it may still be used today to continue with legacy operations. Apple has been pushing for swift by implementing frameworks such as SwiftUI and UIKit to aid with the development of iPhone, iPad, Apple Watch, and Apple TV applications. To launch an application to the app store, the developer needs to pay one hundred dollars yearly and have their application go through a rigorous screen process to ensure quality.

Android development shares numerous similarities to the IOS platform. IT previously used Java as its main programming language (a big advantage as it is the main programming language used by our university) but later converted to Google's Kotlin due to Oracle Corporation's ownership of Java. The Google play store also has a fee to upload applications, but it is much lower than Apple's App Store, currently sitting at thirty dollars. Google's play store also has a screening process to ensure the quality of applications uploaded to their platform.

#### *3.2.13.2 Cross-Platform Development*

Cross-platform mobile application development is used to create one single code base that can be shipped to several mobile platforms. Tools like React, React Native, Xamarin, and Flutter are used to create such applications [12]. The main advantage of cross-platform applications is how these frameworks let smaller teams of developers reach a broader audience without the need to divide resources for two different code bases. A smaller code base with one main programming language creates a faster work environment, making it easy to implement features faster and more efficiently.

The main drawbacks of cross-platform applications are their lack of speed, limited functionality, and limited UX [12]. These applications need an extra abstraction layer and rendering process to possibly convert their cross-platform code into native code, making building and running the application much slower than a native application would.

Cross-platform applications also are on the backfoot of achieving the newer phone functionalities and accessing new APIs due to the slow process of updating their libraries for all platforms. These applications may, unfortunately, lack UX components that are available to the native applications, causing the application to feel less in line with the user experience provided by the device.

## *3.2.15 Camera Data Transmission*

There are two main methods to achieve a connection between the camera and the raspberry pi: through the camera module or a USB port. There are pros and cons to each technique, including lowering the resolution and the lack of compatibility between both devices.

### *3.2.14.1 Camera Module Port*

If our version of the raspberry pi does have a camera module port then the camera module could be considered. There are two versions of camera modules, a standard version, and the NoiR version. The standard version is designed to take pictures and record with normal lighting, while the NoiR version is dedicated to capturing images in the dark with the help of an infrared light source [21]. Connection to both would be fairly simple and python can be used to read from the device.

### *3.2.14.2 USB Webcam*

Webcams usually come with an inferior quality compared to cameras connected to the CSI interface, but webcams are fairly easy to find and the connection is simple through USB. Webcams can be controlled through python packages such as picamera or other command-line applications within the raspberry pi. It is simple to use and set up [21].

## *3.2.16 USB Connection*

### *3.2.15.1 HDMI to HDMI*

HDMI chords are a very common interface for audio/visual sharing. These connections extend to many different sources such as TVs, computers, and any type of monitor. The HDMI cables will be utilized for connection between the Raspberry Pi and the TV monitor that will project the image. Since both the TV and the Raspberry Pi have HDMI access, the extension will be HDMI to HDMI.

### *3.2.15.2 USB to USB*

This project will involve a PCB containing MCU characteristics such like an arduino. The Arduino and Raspberry Pi will need to connect through the USB interface which can be done if the PCB contains a USB port. The Raspberry Pi already contains this.

## *3.2.17 Sensors*

The computer vision task is one of the most computationally intensive tasks in the system, therefore we would not want it to run unnecessarily. The task of a sensor would be to detect the presence of the user and turn the camera on, beginning gesture recognition.

### *3.2.17.1 HC-SR04 Ultrasonic Sensor*

The ultrasonic sensor is a familiar component that we all know about. It uses sonar to detect the distance to an object, much like bats.

| HC-SR04 Ultrasonic Sensor Technical Specifications | |
|---|---|
| Power Supply | +5V DC |
| Quiescent Current | <2mA |
| Working Current | 15mA |
| Effectual Angle | <15 degree |
| Ranging Distance | 2cm – 400 cm/1" 13ft |
| Resolution | 0.3 cm |
| Measuring Angle | 30 degree |
| Trigger Input Pulse width | 10uS |
| Dimension | 45mm x 20mm x 15mm |
| Price | $4.50 |

**Table 17.** *HC-SR04 Specifications*

Some things to note here, the measuring angle is 30 degrees but is most effective within 15 degrees. The ranging distance is from 2 cm - 400 cm. This should be satisfactory for our design as we would assume that the user would most likely enter this radius upon using the device.

The sensor includes four pins Vcc, Trigger, Echo, and Ground. The pin configuration is shown in the table below.

| HC-SR04 Pinout | |
|---|---|
| Vcc | Provides +5V power supply to sensor |
| Trigger | Input pin used to start measurement by transmitting ultrasonic waves by keeping this pin high for 10us |
| Echo | Output pin that goes high for a certain time period. The pulse width corresponds to the distance |
| Ground | Connected to ground of the system |

**Table 18.** *HC-SR04 Pinout*

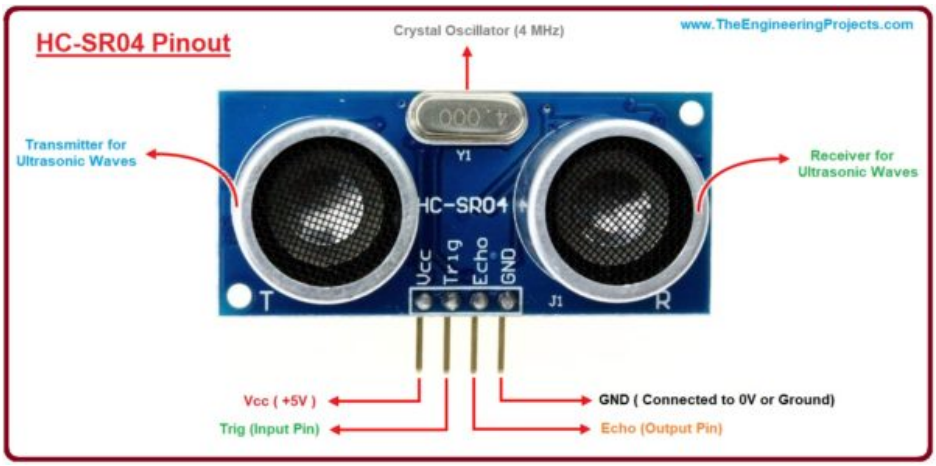


**Figure 28.** ***HC-SR04 Pinout***

### *3.2.17.2 HC-SR501 PIR MOTION Sensor*

The HC -SR501 will detect infrared changes in the environment. The device requires about a minute to initialize and during this time false detection signals are often outputted. The controller logic will have to take this into account. Close by light sources may interfere with the readings of the sensor so that also needs to be taken into account.

| HC-SR501 Technical Specifications | |
|---|---|
| Operating Voltage | 5 -20V DC |
| Quiescent Current | <50uA |
| Level output | High 3.3V / Low 0V |
| Trigger | L position(no repeat)/H (Default repeated trigger) means output will stay high as long as movement in room |
| Delay time | 3-300S( adjustable) Range (approximately 3Sec -5Min) |
| Block time | 2.5S(default)Can be made a range(0.xx to tens of seconds |
| Board Dimensions | 32mm*24mm |
| Angle Sensor | <110 ° cone angle |
| Operation Temp. | -15-+70 degrees |
| Lens size sensor | Diameter:23mm(Default) |
| Price | ~$8.00 |

**Table 19.** *HC-SR501 Specs*

The pinout and controls are as follows.

| HC-SR501 Pinout | |
|---|---|
| Time Delay Adjust | Sets how long the output remains after detecting motions. Ranges from 5 seconds to 5 minutes |
| Sensitivity Adjust | Sets detection range from 3 - 7 meters |
| Trigger Selection Jumper | Sets single or repeatable triggers |
| Ground pin | Connect to system ground |
| Output pin | Low when no motion is detected and hgh when motion is detected. High is 3.3V |
| Power pin | 5 - 20 V DC input |

**Table 20.** *HC-SR501 Pinout*

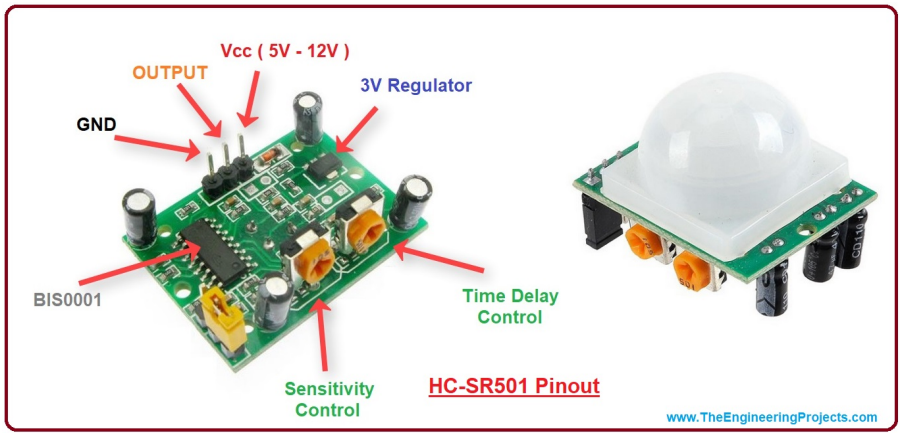


**Figure 27. HC-SR501 Pinout**

### *3.2.17.3 HC-SR505 Mini PIR Motion Sensor*

The HC-SR505 can be considered the little brother of the HC-SR501. Much like the HC-SR505, it is based on infrared technology and detects the motion of a body in front of it. The main feature of this device is its small size. Its small area and low operation mode make it a popular choice in applications where size is a factor and it runs on battery.

| HC-SR505 Technical Specifications | |
|---|---|
| Operating Voltage | 4.5 -20V DC |
| Quiescent Current | <60uA |
| Level output | High 3.3V / Low 0V |
| Trigger | Repeatable Trigger (default) |
| Delay time | Default 8S + -30% |
| Sensing distance | 3 meters |
| Board Dimensions | 40mm * 10mm * 12mm |
| Angle Sensor | <100 degree cone angle |
| Operation Temp. | -20 - +80 degrees |
| Lens size sensor | Diameter: 10mm (default) |
| Price | ~$8.00 |

**Table 21.** *HC-SR505 specs*

The pinout is as shown in the table below.

| HC-SR505 Pinout | |
|---|---|
| Ground pin | Connect to system ground |
| Output pin | Low when no motion is detected and hgh when motion is detected. High is 3.3V |
| Power pin | 5 - 20 V DC input |

**Table 21.** *HC-SR505 pinout*

### *3.2.17.4 Sensor Comparison*

Based on the needs of the project, a PIR motion sensor would be more ideal as compared to an ultrasonic sensor because it senses humans or more specifically the changes in IR light. The ultrasonic sensor would sense anything in its field, including inanimate objects. Since the user will be human the PIR sensors will more reliability give a correct indication that a user is ready to use the device.

The two PIR sensors described in this section are very similar but have some key differences that will be discussed in the rest of this section. The key metrics that need to be taken into consideration will be outlined in the table below.

| Motion Sensor Comparison | | | |
|---|---|---|---|
| **Criteria** | **HC-SR501** | **HC-SR505** | **HC-S04** |
| Operating voltage | 5 - 20V DC | 4.5 - 20V DC | 5 - 20V DC |
| Quiescent Current | <50uA | <60uA | <2mA |
| Adjustable sensitivity | Yes (3-7 meters) | No (3 meters) | No (4 meters) |
| Board Dimensions | 32mm x 24mm | 40mm x 10mm | 45mm x 20mm |
| Adjustable Delay time | Yes | No | No |

| Criteria | HC-SR501 | HC-SR505 | HC-S04 |
|---|---|---|---|
| Level output | High 3.3V / Low 0V | High 3.3V / Low 0V | High 3.3V / Low 0V |
| Price | ~$8.00 | ~$8.00 | |

**Table 22.** *PIR Sensors Comparison*

Both sensors have a similar operating voltage, low current usage, output 3.3V on high, and cost roughly the same. Depending on the vendor. The HC-SR505 does have a slight edge in size, giving it the nice ability to integrate into projects more discretly. The key difference in the comparison is that the HC-SR501 has adjustable sensitivity and delay times via potentiometers on the board. This is ideal to allow for flexibility in our project. The HC-SR501 is also more documented than its counterpart, making it more friendly for first time users. The upsides of the HC-SR501 do make it the better choice in the Holomon.

## *3.3 Strategic Components and Part Selections*

For the hologram component, Pepper's Ghost Illusion will be the primary technology used to create HoloMon. This is because for time and budget constraints, this will allow us to create a proof of concept for our idea within the limitations given and imposed.

There were a lot of options to choose from for the gesture recognition component. Ultimately, we decided to go with the skeleton-based recognition approach since this was the most cost-effective while being practical. The only challenge for this method is range of detection, which the team has decided will just be a design constraint for the camera.

We will also use the Logitech HD Pro Webcam C920 since this component is already acquired making it the most affordable option and has sufficient resolution for the machine learning model to detect a hand in. Also, this camera is USB type A powered which the single board CPU will most likely have many extra USB-A ports to spare.

Since the Logitech HD Pro Webcam C920 is a USB webcam, that will be the main method for camera data transmission to occur. While the camera module with the Raspberry Pi has many features, the USB webcam is a more simplified protocol we can use.

Since there were only two options for the display given the budget constraints and desired results, we decided to use the TV with HDMI connection. This would allow the HoloMon to sit squarely on a table with ample screen size to display the image assets while maximizing the dimensions of the actual hologram.

Either cloud provider would satisfy the product's needs but AWS has a greater community and a more flexible free tier by giving their user's an EC2 instance (shared virtualized environment) for 12 months. The other two cloud providers grant each user a certain amount of free credit to buy these virtualized environments, therefore adding more complexity around their product. AWS also has many other free services that can be utilized whenever without the need to keep track of any initial credits.

As for the backend framework, Python's FastAPI will be used due to its ease of use, high-performance, and python is already used to develop with the raspberry pi and the machine learning algorithm. Python is also an extremely versatile and easy-to-learn language, letting all members of the team contribute and understand the logic behind our server-side application.

React Native would be used as the front-end mobile application framework due to the team's knowledge of JavaScript and its cross-platform capabilities. It is, unfortunately, unsustainable to separate the code base between two mobile platforms due to the size of the team and time constraints, therefore, a cross-platform solution is necessary to cater the application to as many devices

as possible. Flutter was highly considered but its programming language, Dart, is niche.

MongoDB will be used as the database to store user information and maps to the assets. It is not necessary to create a relational schema for this application as the data held is minimal, there should be little joining of data objects, making MongoDB perfectly fine to contain such data. MongoDB can be easily connected to the FastAPI server code as well as it is fairly easy to instantiate within AWS.

As for the housing design, Plexiglass will be used to realize Pepper's Ghost Illusion. The key quality that makes it a good choice is the high light transmittance which will contribute to making the resolution of our hologram higher. It is also relatively cheap, safe, and readily available in stores. Additionally, some wood will be used to house the plexiglass for stability.

The sensor responsible for detecting the presence of an incoming potential user was chosen to be the HC-SR501 PIR motion sensor. It is a well documented and reliable motion sensor that uses infrared emitted by human bodies to detect presence. The distinguishable features that made it suitable for our project was the ability to adjust sensitivity (detection range) and delay time(how long the output will remain high once motion is detected).

The main power sources being used for this project will be wall outlets for the single board CPU and the display unit while a 9V battery will be used to power the microcontroller.

The microcontroller unit selected for HoloMon is an Arduino Uno since it was one of the microcontrollers the team had readily available while also providing enough SRAM and flash memory for the project while being the cheapest option.

The single board CPU chosen for this project is a Raspberry Pi 4 Model B since it was also one of the single board CPUs the team had readily available. This single board CPU was selected because many of the members already had prior familiarity with coding a Raspberry Pi while there are many resources on the Internet for pairing gesture recognition machine learning models to it as well.

## *3.4 Part Selection Summary*

Below is the summarized table showing all the selected components from each part that is necessary to the project.

In summary the Raspberry Pi Rev 4 will be the backbone of the project leading as the computer. Comparing all the Revisions of the Raspberry Pi’s to each other, it was clear to see that the latest Revision was best suitable for this project. The Raspberry Pi was also compared to other Computer Boards and although other options would have been suitable, since the Raspberry Pi was already obtained it still remained the best option. The PCB board may not contain all of

the Arduino capabilities but it will have the most important parts: the chip and the pins. Since it is not a manufactured Arduino, certain parts of the PCB may have to be incorporated later such as the CP2102 converter that acts as the USB for the PCB. This part is important for Raspberry Pi and Ardunino connection interface.

The Pi's power source is a LABISTS plug head which will be taking power from the wall. The microcontroller chosen will be the Arduino Uno. This MCU will be chosen distributed onto a PCB board and will have Ardunino like features. The power source for the Arduino will be a power jack that will connect to a 9V battery for independent uses. However for the project itself, the PCB will be powered by the Raspberry Pi which is powered by the wall. This option was the simplest and easiest when working on the PCB design and will leave room for less wires on the project and less complications. Details of the PCB prototype will be discussed in chapter 6. The TV monitor, like the Raspberry Pi will be plugged into the wall outlet. The camera will be powered by the Raspberry Pi via USB.

HDMI and USB cables have been selected to interface from the Raspberry Pi to the TV and from the Raspberry pi to the Arduino on the PCB. The gesture recognition will be camera-vision based and the camera chosen is the Logitech HD Pro Webcam C920 since it is USB-A powered, fitting with the Raspberry Pi and has an easier camera data transmission method.

The HC-SR501 PIR Motion sensor was selected for this project to ensure that the camera that will be detecting motion sensing will not be on and wasting power when the user is not interacting with the HoloMon. Certain things can be unplugged such as the TV and the Raspberry Pi, however, since the camera will be functioning so long as the Raspberry Pi is functioning, the Webcam will be using a lot of unnecessary power. The Sensor will be utilized to turn off the Webcam when there is no detected motion. This most likely will occur with the communication interface of the Raspberry Pi shutting off the webcam when the Arduino/PCB sends data to the Raspberry Pi that the motion sensor has not detected any motions/hand gestures in about 5-8 minutes. When the sensor detects the motion, the Webcam will immediately turn on. The specific HC-SR501 sensor is extremely simple and easy to use with the Arduino Rev 3. There are a total of 3 connections that need to be used for this component, the power, ground, and a pin.

Pepper's Ghost Illusion was selected because it was an easy simple way to create a 3D image to be projected. This Illusion has been around for decades and been utilized in many theme parks. There are a couple constraints with this such as not being able to see the hologram in certain positions. Another constraint is that making this illusion requires space and a large monitor. So this HoloMon won't be able to be mobile and will most likely be bulky which is not ideal for younger users which is the key audience. However, since the electronic devices will need to be connected to the wall outlet, this will nicely correlate with the immobile HoloMon. This illusion is perfect to ensure a more simple design in the base structure of the Hologram.

| Device Compared | Final Selection |
|---|---|
| **Single Board Computer** | Raspberry Pi 4B |
| **Microcontroller** | Arduino Uno Rev3 |
| **Power Source** | LABISTS, Wall Outlet, 9V battery |
| **Database** | MongoDB |
| **Cloud Service** | AWS |
| **Hologram** | Pepper's Ghost Illusion |
| **API** | FastAPI |
| **Front End** | React Native |
| **Display** | TV |
| **Camera** | Logitech HD Pro Webcam C920 |
| **Sensor** | HC-SR501 PIR Motion Sensor |
| **Connection Cables** | HDMI to micro-HDMI, USB to USB |
| **Housing Unit** | Plexiglass, Wood |

**Table 23.** ***Part Selection Summary***

After extensive research into the major and minor components involved in creating this project, the final selection of said materials is listed in the above table.

# 4. Related Standards and Realistic Design Constraints

The following section, Related Standards and Realistic Design Constraints, outlines the standards for different practices used in our project and the design impact of relevant standards along with economic and time constraints, environmental, social, and political constraints, ethical, health, and safety constraints, and manufacturability and sustainability constraints.

## *4.1 Standards*

The Standards subsection details the standards for python coding, React native coding, mongoDB coding, and C programming styling. A hardware specific standard is outlined below for PCB manufacturing.

### *4.1.1 PEP8 - Style Guide for Python Code*

The PEP8 styling and coding standards are a set of guidelines on writing Python code initially written in 2001 by Guido van Rossum, Barry Warsaw, and Nick Coghlan [5]. The document that outlines the conventions evolve over time as some become obsolete and other changes in the language become more commonplace. The major theme behind this styling guide is knowing that “readability counts,” and that “consistency is key.” However, it is still a guideline, and there are moments where applying PEP8 would make the code less readable or make a product not backwards compatible.

The first section speaks to indentation. There should be 4 spaces per indentation level. Lines that continue should align wrapped elements by using Python’s implicit line joining inside parentheses, brackets and braces, or using a hanging indent. The 4-space rule is optional for continuation lines. There is no explicit position on how (or whether) to further visually distinguish conditional lines from the nested suite inside an if-statement. The closing brace/bracket/parenthesis on multi-line constructs may either line up under the first non-whitespace character of the last line of a list or may be lined up under the first character of the line that starts the multi-line construct. Spaces are the preferred indentation method. Tabs should only be used to remain consistent with code that is already indented with tabs.

The next section discusses maximum line length. Limit all lines to a maximum of 79 characters. For flowing long blocks of text with fewer structural restrictions such as docstrings or comments, the line length should be limited to 72 characters. In PEP8, it is permissible to break before or after a binary operator as long as the convention is consistent locally. For new code, Knuth's style is suggested [5]. However, using line breaks after a binary operator tends to make it more readable. For blank lines, try to surround top-level functions and class definitions with two blank lines. Method definitions inside a class are surrounded by a single blank line. Extra blank lines may be used sparingly to separate groups of related functions. Blank lines can be omitted between a bunch of related one-liners. It's a good idea to use blank lines in functions, sparingly, to indicate logical sections.

For source file encoding, code in the core Python distribution should always use UTF-8 (or ASCII in Python 2, although this version is typically outdated by now). Non-default encodings should be used only for test purposes or when a comment or docstring needs to mention an author name that contains non-ASCII characters; otherwise using \x, \u, \U, or \N escapes is the preferred way to include non-ASCII data in string literals. All identifiers in the Python standard library must use ASCII-only identifiers, and should use English words wherever feasible. Also, string literals and comments must also be in ASCII.

When it comes to imports, according to PEP8 styling guide, they should be on separate lines unless importing multiple modules from a single library. They are always put at the top of the file just after any module comments and docstrings and before module globals and constants. Imports should be grouped in the following order: 1) standard library imports, 2) related third party imports, and 3) local application/library specific imports. There should be a blank line between each group of imports. Absolute imports are recommended unless they become unnecessarily verbose.Implicit relative imports should never be used and have been removed from Python 3 and later versions. In regards to module level dunder names (double underscore names, i.e. __all__) should be placed after the module docstring but before any import statements except from __future__ imports. Python requires that future-imports must appear in the module before any other code except docstrings. PEP8 does not make a recommendation for string quotes, whether they are single or double quote characters. However, for triple-quoted strings, use double quote characters to improve readability.

PEP8 has some standards regarding whitespace in expressions and statements. Avoid extraneous whitespace in the following situations: immediately inside parentheses, brackets or braces, between a trailing comma and a following close parenthesis, and immediately before a comma, semicolon, or a colon. But it is okay to have whitespace wherein the colon acts as a binary operator and should have equal amounts on either side.

In regards to comments, it should always be a priority to keep the comments updated when the code changes. They should be in complete sentences. Its first word should be capitalized, unless it is an identifier that begins with a lowercase

letter. If a comment is short, the period can be omitted. Use two spaces after a sentence-ending period. Each line of a block comment starts with a # and a single space unless it is indented text inside the comment. Paragraphs inside a block comment are separated by a line containing a single #. Inline comments should be used sparingly. They should be separated by at least two spaces from the statement and start with a # followed by a single space. Write docstrings for all public modules, functions, classes, and methods [5]. They are not necessary for non-public methods, but comments should describe what that method does.

Naming conventions are still undergoing standardization but the following are recommended guides. When using abbreviations in camel case, capitalize all the letters of the abbreviation. Never use ‘l’ (lowercase el), ‘O’ (uppercase oh), or ‘I’ (uppercase eye) as single character variable names. Package and module names should have short, all-lowercase names. Underscores may be used in the module name if it improves readability, though discouraged. Class names should be written in camel case convention. Functions may be used instead in cases where the interface is documented and used primarily as a callable. Type variables should use camel case preferring short names. Function names should be lowercase, with words separated by underscores as necessary to improve readability, unless mixed case is the dominating style. Always use ‘self’ for the first argument to instance methods. Always use ‘cls’ for the first argument to class methods. If a function argument’s name is the same as a reserved keyword, append a trailing underscore over using abbreviations or intentional misspelling. Constants are defined in all capital letters with underscores separating words.

### *4.1.2 React Native Coding Standards*

Unlike Python with PEP8, a tried and true styling guide for React Native does not exist. Gilshaan Jabbar in [7] put together documentation by using references from React Native documentation, Medium documents, and his experience in coding. Naming conventions in React Native are different. A folder and sub folder name should always start with small letters and the files belonging in the folders are in pascal case. Pascal case can be described as any compound word where the first letter of each word is capitalized, i.e. HoloMon. Component names should also follow the pascal case. Files in folders with the same name should be avoided. There should be no space between two imports. Class names should be declared as file names that will be easy to import and maintain standard declaration. The object and variable declaration should always be in camel case statement. If using semicolon, then  use in all places at the end of statements or do not use.

React Native is unique in that like other front end development languages, requires a very organized file system structure. All components, globals, images, redux, etc. should be written inside the ‘app’ folder [7]. All components except global components should be written inside the ‘components’ folder under the

‘app’ folder, while global components should be written inside the ‘global’ folder under the ‘app’ folder. Images should be in the ‘images’ folder under the ‘app’ folder. Global functions for API requests should be written in the ‘request’ folder under the ‘app’ folder. If using redux, then the redux files should be written inside the ‘store’ folder under the ‘app’ folder. The localization file is directly written in the ‘app’ folder.

Imports in React Native should follow this order all sorted by alphabetical order if necessary: 1) React import, 2) Library imports, 3) absolute imports from the project, 4) relative imports, 5) import * as, and 6) import ‘./<some file>.<some extension>.’ Each group should be separated by an empty line.

Layout conventions in React Native follow other conventional coding styles relevant to JavaScript. Always end a statement with a semicolon. Create a class component if using state, otherwise use a functional component. Do not allow a state to be invoked on Render() of a React component. Indent continuation lines if they are not indented automatically with one tab stop (four spaces). Add at least one blank line between method and property definitions. There should be no line space between two similar looking statements or similar groups of coding applied to the same activity [7].

There are some similarities for commenting conventions between React Native and Python. Place the comment on a separate line, not at the end of a line of code. Begin comment text with an uppercase letter. End comment text with a period. Insert one space between the comment delimiter and the comment text. Attach comments to code only where necessary, to keep the code visually clutter free and avoiding potential conflict between comment and code if the comment is not updated when the code is changed.

React Native is a little more forgiving when it comes to standards in the language. Use any type of naming convention for variables, but stick to one convention throughout. In React Native there are 3 types of quotes. Double quotes, single quotes, and backticks. There is no difference between them in React Native. Backticks are “extended functionality” quotes. Avoid repeating the same piece of code twice by creating reusable components. Put stylings in a CSS stylesheet instead of inline stylings so the code only needs to change in one place. Exception handling plays an important role in mobile and web development. Using try and catch blocks to handle exceptions is standard within a React Native app. Put all API calls in componentDidMount() since this will make it clear that data will not be loaded until after the initial render. This standard assists in setting up the initial state properly to avoid having an undefined state that results in errors. If using react native hooks, write api calls within the useEffect.

Some standards that pertain to mobile and web development that may not be React Native exclusive include always performing both local and server validations. Always implement server validations and client validations where possible. Ensure that the app is responsive, meaning it is consistent across different devices and platforms. Add loading indicators to make the app look

more responsive and professional while fetching data or waiting for an API response. Avoid putting logs in the release build. Disable all logs when doing a release build via manually or with a script. With newer devices, React Native has a nifty feature called SafeAreaView which provides automatic padding when a view intersects with non-screen material, i.e. a notch, status bar, or home buttons. Similarly, use keyboard avoiding view, so that the app can automatically adjust its height, position, or bottom padding based on the keyboard height of the mobile device. Follow strict linting rules. Remember the difference between pushing the screen and navigating the screen. Navigating the screen is using a route only once and should not come up again when committing actions like the back action. Pushing the screen loads the same component twice with different data, and will be able to return to the previous screen with the back action.

### *4.1.3 MongoDB Standards and Guidelines*

The overarching goal for MongoDB and their style guide is to “bring uniformity, avoid ambiguity, and more importantly have seamless integration of information across the enterprise,” [8]. The following naming standards are for MongoDB database naming, collection naming, and field naming convention.

For database naming, use camel case or lowercase as required. Use alpha-numeric characters only. Recall that database names are case-sensitive, which may make lowercase preferred. Database names must have fewer than 64 characters. There are variations of naming conventions for database names based on the operating system. The only difference is that Linux/Unix allows for more special symbols than Windows. Database names cannot be empty. They should not contain null or space characters or an empty string. Try to use a delimiter to make the name more readable, such as the underscore or hyphen.

For collection naming, use camel case or lowercase naming conventions. Similarly, lowercase is preferred to avoid case sensitivity. Collection names should not begin with special characters. Once again, try to use some delimiter to make the name more readable. The maximum length of collection namespace should be equal to or less than 120 bytes.

Field naming conventions for MongoDB follow similar patterns to the above two. camel case or pascal case or lowercase. Field names cannot contain the null character or empty string or ‘.’. Field names cannot start with ‘$’. Avoid using separators for the field names, especially if creating/automating model entities from JSON schema. Sometimes delimiters such as the hyphen or underscore might flag as coding issues if mapped to code directly [8].

### *4.1.4 C Programming Standard*

Carnegie Mellon University (CMU) developed a standard that was adapted from a previous C++ they made and NetBSD’s styling guidelines.

The first section goes over naming conventions. The guide talks about the history of names in general and seeks to instill some accountability to “make names fit,” [28]. As a convention stated in [28], “if the name is appropriate everything fits together naturally, relationships are clear, meaning is derivable, and reasoning from common human expectations works as expected.” CMU affirms that if all the names in a program can be things like “foo” and “bar” that it is best practice to revisit the design. In general, every function performs some kind of action, so the name should be clear in its function, i.e. check_for_errors() is better than error_check() and dump_data_to_file() is better than data_file(). Following this pattern of naming functions with verbs first will distinguish itself from data objects. Since structs are often nouns, they can be named as such and therefore be easily identifiable when compared to functions. Appending prefixes or suffixes are sometimes useful. Some examples of prefixes include ‘is’, ‘get’, ‘set’ while some examples of suffixes include ‘max’, ‘cnt’, and ‘key’. Using ‘is’ may be useful for setting booleans since it means to ask a question about something. If a variable represents time, weight, or some other measurable unit, then it is best to include the unit in the name for easier debugging. Two examples are variables such as timeout_msecs and my_weight_lbs. For structure names specifically, there is an order in how it should be named: firstly by use, then by size, and lastly by alphabetical order. The manner here is an attempt to minimize memory waste because of compiler alignment issues. The example given is that using “int a; int b; char *c; char *d” is better than “int a; char *b; int c; char *d.” Another practice is that each variable gets its own type and line although exceptions can be made when declaring bitfields. However, using bitfields in general is discouraged in C coding. “Major structures should be declared at the top of the file in which they are used, or in separate header files, if they are used in multiple source files,” [28]. Separate declarations should be made for use of the structures and include an “extern” if they are declared in a header file. For variable names on the stack, using all lowercase letters and ‘_’ as the word separator is appropriate. The reason for this is so that the scope of the variable is clear and that all variables will look different and become easily identifiable in the code. When using pointer variables, place the ‘*’ close to the variable name not pointer type, i.e. char *name is better than char* name. Global variables should be prefixed with a ‘g_’ and should altogether be avoided when possible because it is important to know the scope of a variable. Global constants should be declared using all caps with ‘_’ separators because it is tradition to have them named this way. However, CMU cautions against having the constants conflict with other global #defines and enum labels when styling it this way. Macro names and #defines should also be in all uppercase using ‘_’ separators. Macros are capitalized, parenthesized, and should avoid side-effects. Spacing before and after the macro name may be any type of whitespace though keep the use of tabs consistent throughout a file. If a macro name is an inline expansion of a function, the function remains defined in all lowercase, and the macro has the same name but all in uppercase. If the macro is an expression, wrap the expression in parenthesis. If the macro is more than a single statement, use a do-while loop so that a trailing semicolon works. Make sure to right-justify the backslashes because it makes it easier to read. The

justification behind these styling guidelines is so that it becomes very clear that the value is not alterable and for macros, it makes it clear that the developer is using a construct that requires attention. Also, some hard to catch errors may occur when macro names and enum labels use the same name. Enum names should be expressed in the same way as macros and #defines using all caps with '_' separators since this is the standard rule for enum labels. No comma should be present on the last element. It is best practice to also make a label for an error state such that if a label becomes uninitialized or moved into an error state, the state machine will know where to go. Make this the first label if possible.

The second section defines styling for formatting. CMU first goes over when braces are needed. "All if, while, and do statements must either have braces or be on a single line," [28]. This is to ensure that when someone adds a line of code later there are already braces they will not forget while also providing a more consistent look. Furthermore it is easy to do and does not affect execution speed. Alternatively if the code can be placed in a single line, it provides safety when adding new lines since the developer will know it is only a single statement while maintaining a compact readable form. Adding comments to closing braces can help with code tracing so that at a glance a developer does not have to find the beginning brace to understand everything. Consider screen size limits and keep it to the width of a common screen size so that scrolling horizontally to read code is not necessary. Parens () with key words and function policy is mentioned next. It is standard to not put parens next to keywords but rather a space between. Also do not put parens next to function names. Lastly, do not use parens in return statements when not necessary. This is because keywords are not functions and putting parens next to keywords will make keywords and function names look identical. Max line size is 78 characters. If then else formatting is left up to the programmer. CMU recommends, however, to always have an else block for finding unhandled cases if the tree requires else if statements or even putting a log message in the else if no corrective action is declared. Always put the constant on the left hand side of an equality/inequality comparison, i.e. if (6 == errorNum) … since this will allow the compiler to find the error if the programmer misses one of the equal symbols. Another reason is that it puts the value at the front instead of buried at the end of an expression. For switch formatting, put comments in between the next case statement if any. The default case should always be present and trigger an error if it should not be reached but is reached. If variables are necessary, put all of the code in a block. CMU also has a standard for using goto, continue, break, and ?: internal functions. Goto statements should be used sparingly, but if it is necessary, the accompanying label should be alone on a line and to the left of the code that follows. Additionally, the goto should be commented (possibly in the block header) as to its utility and purpose. Continue and break is similar to gotos and should be used sparingly because it may bypass the test condition and or bypass the increment/decrement expression. A further rule that may be given is do not mix continue with break in the same loop. If a ternary operator such as ?: works well, the following rules would help to standardize them: put the condition in

parens so as to set it off from other code, if possible, the actions for the test should be simple functions, and lastly put the action for the then and else statement on a separate line unless it can be clearly put on one line. The next section elaborates on the one statement per line standard and lists more reasons why it is good practice. The reasons are that the code is easier to read with white space. One variable per line is good practice because documentation can be added for the singular variable on the line and it becomes clear that the variables are initialized and that declarations are clear and reduces the possibility of accidentally declaring one data type over another [28]. Generally speaking, enums as constants should be deprecated thanks to C programming's ability to allow for constant variables. However, this is not always a desired feature since in most compilers the constants will take space. So it is better to use enums in the case of tight memory constraints like embedded systems, however all other types of programmers should continue with the rest of the discussion and that in general enums are preferred to #define as enums are understood by the debugger. However, enums are not given a guaranteed size and so it may be better in that case to use constants or #define. Never cast between integers and enums. Use header file guards such as #ifndef … #endif to protect against multiple inclusion through the use of macros. The next subsection of formatting describes macros. CMU cautions against turning C into Pascal by macro substitution. Replace macros with inline functions. However, macros for small functions are ok. Macros should be used with caution because of the potential for error when invoked with an expression that has side effects. Always wrap the expression in parenthesis. This avoids potential commutative operation ambiguity in the macros. Similar to global variables, macros can conflict with macros from other packages so try to prepend macro names with package names and avoid simple and common names like MAX and MIN. Make sure to initialize all variables because more problems can occur because a pointer or variable were left uninitialized. Make user-defined functions short. Functions should limit themselves to a single page of code. The idea is that each method represents a technique for achieving a single objective [28]. Most arguments of inefficiency turn out to be false in the long run and that true function calls are slower than not. Always document a null body for a for or while statement so that it is clear that the null body is intentional and not missing code [28]. Do not default if test to non-zero. For example if (FAIL != f()) is better than if (f()) because although they are the same in C which will return false, an explicit test will help to debug later. This method of explicit comparison should be used even if the comparison value will never change as to reflect the numeric and not boolean nature of the test. Generally avoid embedded assignments such as d = (a = b + c) + r; and instead write them on separate lines. Despite the former saving one cycle, the time difference between the two will decrease as the optimizer gains maturity while the difference in ease of maintenance will increase as human memory of understanding with the latter begins to fade.

The next section is a standard on documentation. Consider comments to be a story describing the system that will inform someone else at another point in time just exactly what happened and why. Comments should document decisions [28].

At each point where a choice was made, place a comment describing which choice and why. Use headers structured in such a way that they can be parsed and extracted by using a document extraction system like Doxygen. If done right, no more documentation may be necessary. Each part of the code has a specific comment layout, please refer to Doxygen for further information. If using gotchas, explicitly comment variables changed out of the normal control flow or other code that is likely to break during maintenance. For formatting gotcha, make the gotcha keyword the first symbol in the comment. Comments inside may consist of multiple lines, but the first line should be a self-containing meaningful summary. The writer's name and the date of the remark should be part of the comment. Oftentimes gotchas stick around longer than they should. Embedding the when and who information allows future developers to find out who to direct their questions to. Function headers should be in the file where they are declared. This means that most likely the functions will have a header in the .h file. However, functions like main() with no explicit prototype declaration in the .h file should have a header in the .c file [28]. Include statement documentation should be also considered because it tells the user why a particular file was included.

The next section is omitted called Layering as it does not have any influential design decisions for this project in particular. Skipping to the next section, CMU provides general advice about C programming standards. Do not use floating-point variables where discrete values are needed. Using a float for a loop counter is bad. Always test floating-point numbers as <= or >= but never use an exact comparison. All compilers have bugs. Write "around" compiler bugs only when forced to use a particular buggy compiler. Do not rely on automatic beautifiers. They can only be applied to complete, syntactically correct programs and are not available when the need for attention to white space and indentation are greatest. Programmers, following the above standards, can do a better job of making the complete visual layout clear. Use const in the right places so that parameters that are not meant to be changed in a method do not get changed. Use #if and not #ifdef for macros. When commenting out large code blocks for testing, use #if 0 and #endif to wrap the block instead of /**/ because comments can not contain comments. Do not use #ifdef because someone later may unknowingly trigger ifdefs from the compiler command line. Do not put data definitions in header files. As standard practice, do not use magic numbers. These are considered magic if no one including the author after 3 months knows where the number came from. Lastly, CMU addresses error return check policy. Check every system call for an error return unless you are deliberately ignoring errors. Include the system error text for every system error message [28]. Check every call to malloc or realloc so that other developers do not have to make memory checks everywhere. Using a new might be helpful in a wrapper for these cases.

### *4.1.5 PCB Manufacturing Standard*

There is an organization formally called the Institute of Printed Circuits (IPC) but is presently called the Association Connecting Electronics Industries despite continuing the PIC moniker. IPC is internationally recognized for their PCB-related standards and have since become the electronics-industry-adopted standards for design, PCB manufacturing, and electronic assembly [29]. IPC standards help to ensure quality, reliability, and consistency in electronics manufacturing. The following section outlines the IPC-2221 document called Generic Standard on Printed Board Design. This document acts as the umbrella specifications that led to other standards to be developed for subspecific topics such as discrete wire, rigid, and flex, to name a few. All aspects and details of the design requirements are addressed to the extent that is possible to apply to a broad spectrum of using organic materials or organic materials in combination with inorganic materials (metal, glass, ceramic, etc.) to provide the structure for mounting “electronic, electromechanical, and mechanical components,” [29]. It is intended that after part selection, the user should obtain the sectional document specific to the chosen technology for further standards. This generic standard applies for components that may be through-hole, surface mount, fine pitch, ultra-fine pitch, array mounting or unpackaged bare die. The materials may be any combination that allows the product to perform their respective physical, thermal, environmental, and electronic function. Users of the IPC-2221 guidelines are expected to use metric dimensions. Designing the features and selecting materials for a PCB involves balancing “the electrical, mechanical and thermal performance as well as the reliability, manufacturing and cost of the board,” [29]. The end product requirements should be known prior to design start-up. Maintenance and serviceability requirements are important factors that need to be addressed in the design phase. These factors have a tendency to affect layout and conductor routing.

The next section has a brief section on the parts list, which is a chart comparison of materials used in the construction of PCB assembly. All end item identifiable parts and materials should be identified in the parts list or on the field of the drawing. All mechanical parts should be assigned an item number when appearing on the assembly pictorial and they should match the item number assigned on the parts list. Electrical components, for example, resistors, capacitors, fuses, etc., should be assigned reference designators. The assignment of electrical reference designators should be the same as those assignments given to the same components on the Logic/schematic diagram.

Under test requirement considerations, a review meeting should be held with fabrication, assembly, and testing [29]. Things to review include circuit visibility, density, operation, circuit controllability, partitioning, and special test requirements. For the assembly testability, it is imperative to test the compatibility with the overall integrations, testing, and maintenance plans.

Many of the sections in the IPC-2221 are irrelevant to the scope of this project, however, some important notes are summarized and included to utilize the

pieces of the standards that do apply. 3.6.1 Board Layout Design is one such section where they have standards of the layout of the PCB. There should be designated areas that are identified by function, i.e. power supply section confined to one area, analog circuits to another, and logic circuits to even another [29]. Doing this would help minimize crosstalk and simplify bare board and assembly text fixture design while also facilitating easier troubleshooting diagnostics. The physical parameters of the PCB should be consistent with the mechanical requirements of the electronic system. The decision for board type, whether single-sided, double-sided, multilayer, metal core, etc. should be made prior to starting layout procedures and be based on assembly performance requirements, heat dissipation, mechanical rigidity requirements, electronic performance, and anticipated circuit density [29]. Boards should be of a uniform size.

## *4.2 Realistic Design Constraints*

Currently there are a few constraints with the design idea to prevent too much complexity in the project. The first is that the human-following robot will follow any human instead of a specific human. This is largely due to the fact that we would need to introduce some kind of device pairing service between the robot and most likely a phone, making HoloMon more difficult to create in the time frame given. Another constraint is that the hologram can only be changed by a single person at a time. Since we are creating a mobile application that can change the appearance of the HoloMon, it would be out of scope to introduce the ability for two separate instances of the mobile application to control the hologram simultaneously.

Currently, there are only 10 gestures picked up by the pre-trained, open-source machine learning application. Users would only have 10 ways to interact with the HoloMon by this method and collecting more data would be infeasible due to the time constraint and budget.

Assets may also be troublesome since a variety would be needed to provide a decent user experience. The resolution would be minimal due to the number of assets to be created and due to the lacking technology to create holograms within our budget.

Another constraint considering the current project set up is that the chosen Holographic image is not 3D. This means that the owner of the HoloMon will not be able to see the image of their character unless it is adjusted to the user's POV. Though this constraint does make the project a little less advanced, it is the cheapest and most efficient way to create the HoloMon. This constraint is easily fixed utilizing mirrors for the owner to adjust.

Continued research into our project focus also suggests that holograms, if using more available (and thereby cheaper methods) of displaying the animation, will be susceptible to vibrations and the amount of light in a given room.

If using Pepper's Ghost illusion, the hologram will only be visible in indoor settings.

One constraint to bring to obvious attention is difficulties with gesture recognition in the dark. Since the HoloMon is essentially made up with light the HoloMon is to interact with the user in darker settings for clarity. However, doing so may make it a lot more difficult for the user to interact with the HoloMon since the hologram is interfaced through a camera.

This can be controlled or modified by incorporating some kind of camera that has decent night vision. Another way to ensure that this does not happen is possibly adding some kind of light to the camera so the camera can see the user but also the user will be able to still see the Hologram.

Also, two more design constraints include the requirement for WiFi and Bluetooth to be present.

### *4.2.2 Environmental, Social, and Political Constraints*

The main problem with this design would be the abundant amount of hardware needed to create the product. Applications that rely on hardware are major contributors to E-Waste. Electronic waste, often known as e-waste, refers to gadgets that have been dumped electrically or electronically. E-waste includes used electronics that are meant for refurbishment, reuse, resale, salvage recycling through material recovery, or disposal. In underdeveloped nations, informal e-waste processing can have negative health and environmental consequences [43].

Lead, cadmium, beryllium, and brominated flame retardants are found in electronic trash components such as CPUs, which can be dangerous. The health of workers and their communities may be jeopardized by the recycling and disposal of e-waste [43].

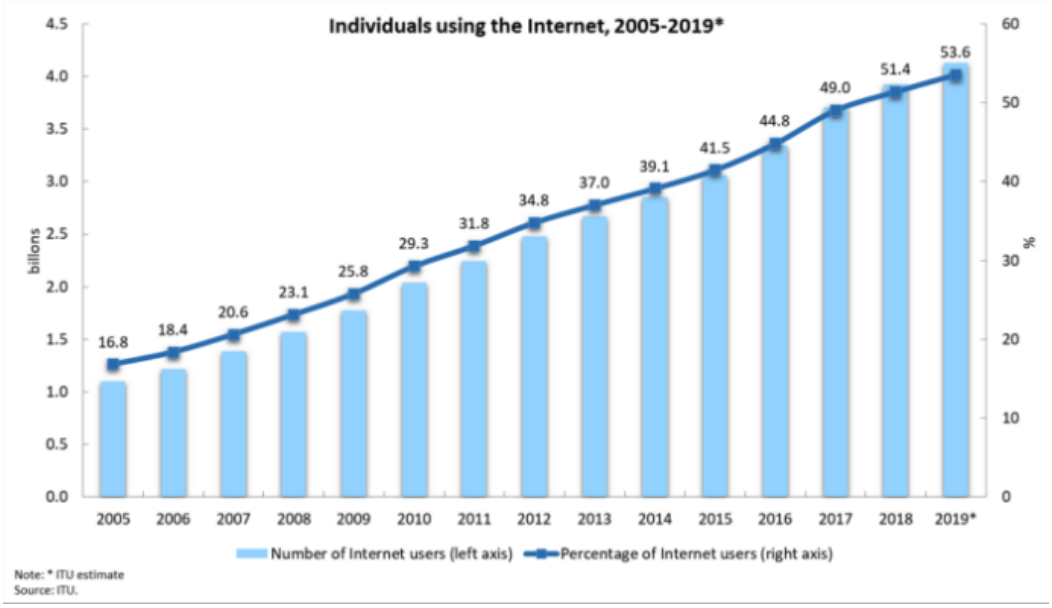


**Figure 29,** *Growth of E-Waste*

## *4.2.3 Ethical, Health, and Safety Constraints*

When evaluating whether a product should be developed it's important to analyze the product's environmental impact and or possible safety constraints. When developing the HoloMon, there are about 4 constraints:

- Possible harm from the weight of the HoloMon
- Sharp edges and or small spaces in the Hologram
- Flashing lights
- Eye strain

### *4.2.3.1 Weight from Machinery*

One safety constraint of the HoloMon is possibly the size of the Hologram itself. Since this is a toy for younger children, it may be a little dangerous depending on the age of the user. The Hologram's display is estimated to be about 33 inches (possibly more). Depending on how heavy the Hologram is, if it somehow falls off a desk, or if the desk falls, this could possibly cause harm to anyone near it since it will be a decently heavy machine.

However, even with this constraint, it must be noted that the Hologram will not be mobile and any risk of it falling off the desk and or platform will be due to an unstable platform and or the user's fault. There will be an age

limit as to who recommended these devices. Toddlers will certainly not be recommended, but kids older than 9 will be approved.

#### *4.2.3.2 Sharp Edges*

Another small safety constraint is the possibility of sharp edges and small spaces in the Hologram Machine. Since the design has not been fully developed, the possibility of having sharp edges and species for users to get their fingers stuck is high. However, this project will try to eliminate the possibilities for potential harm.

#### *4.2.3.3 Flashing Lights*

One other possibility of potential harm is if the user has Photosensitive epilepsy. Due to the lasers/lights projecting the hologram, there are many potential flashing lights that could affect someone sensitive to flashing lights. This is a major concern and there will be a warning for those with PSE. Especially since the Hologram is interactive and changing which could create a quick flashing light harmful to those who have PSE.

#### *4.2.3.4 Eye Strain*

Another constraint is possible eye strain for any user. Since the hologram is better seen in darker areas, it is possible that staring at the Hologram for too long can strain the user's eye. Just like staring at a phone screen for too long in the dark can hurt your eyes, so can this hologram since intense lights will be used to project the HoloMon. Also with the possibility of the camera including a light to see the user, as referred to in constraints, there is a large possibility of the user's eyes being strained. However this route is most likely not going to be chosen for the camera due to the impact it will have on the user's eyes.

### *4.2.4 Economics and Time Constraints*

In this subsection, the project, like any other, is subjected to economics and time constraints that the team has taken precautions to deal with and prevent. Such measures include using already acquired electrical components, purchasing any with haste after selection, and narrowing the scope of the prototype to be built.

#### *4.2.4.1 Semiconductor Shortage*

Semiconductor devices are devices that are able to be a conductor and or and insulator depending on certain conditions. The physics behind a semiconductor basically enables a circuit to control the conduction current inside the semiconductor. Controlling this current is useful in almost every electronic device. There is a gap between the insulator and the conductor called the band gap that demonstrates how much energy it will take for an electron to leave and enter the conduction band. Whenever someone

would want to overcome the bandgap they would just use a power source to enable electrons to be added and more bonds to be recombined. There are many different ways to get a semiconductor to shift from a P-type in the semiconductor to an N-Type. The most popular and affordable way is to increase the amount of impurities. This method is called doping and is used to shift a junction from a positively charged semiconductor to a negative.

Semiconductors are the backbone of every single electrical device. Computers, phones, ATMs, and trains are all integrated through the semiconductor devices. How exactly does this affect the HoloMon? Semiconductors are basically any type of material such as Silicon that is the platform for a chip(CPUs). CPUs run an entire device such as Raspberry Pi's, Arduinos, and any other device that contains a computer-like chip.

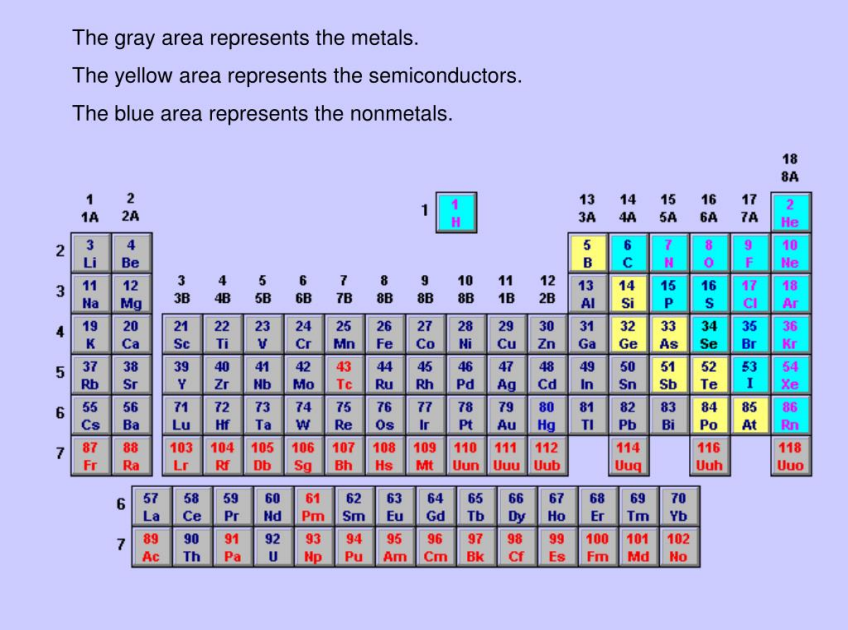


**Figure 30.** *Semiconductor Elements (Yellow)*

Due to the pandemic, a semiconductor chip shortage has affected pricing and postponed any purchases. The shortfall will not be completely fixed for a while [46]. One thing sparking the shortage is that the tiny semiconductor chip was a huge function for cars as more smart cars were being developed. Since smart cars have many functions that need to be basically operated by a semiconductor.

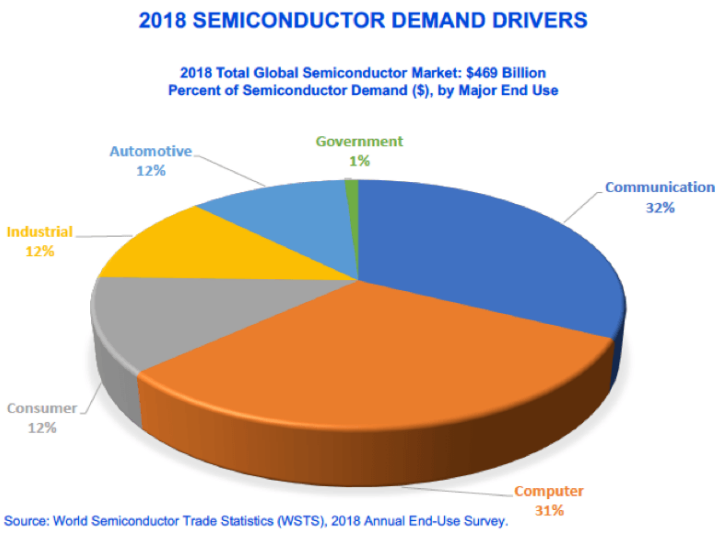


**Figure 31.** *Semiconductor Types of Devices 2018*

Since it's in high demand, the supply chain has been having difficulty supporting all the demands. One leading factor to the shortage of semiconductors is the “insufficient capacity at semiconductor fabs” [46]. Since car sales had been decreasing in 2020, the industry adjusted to selling less semiconductors, but then with the spike of demand for automobiles, the suppliers have yet to balance back. The government has tried to incorporate and delegate investments in semiconductors to help shortages. Since semiconductors are so hard to acquire, the possibility of solving this shortage is not likely to be soon.

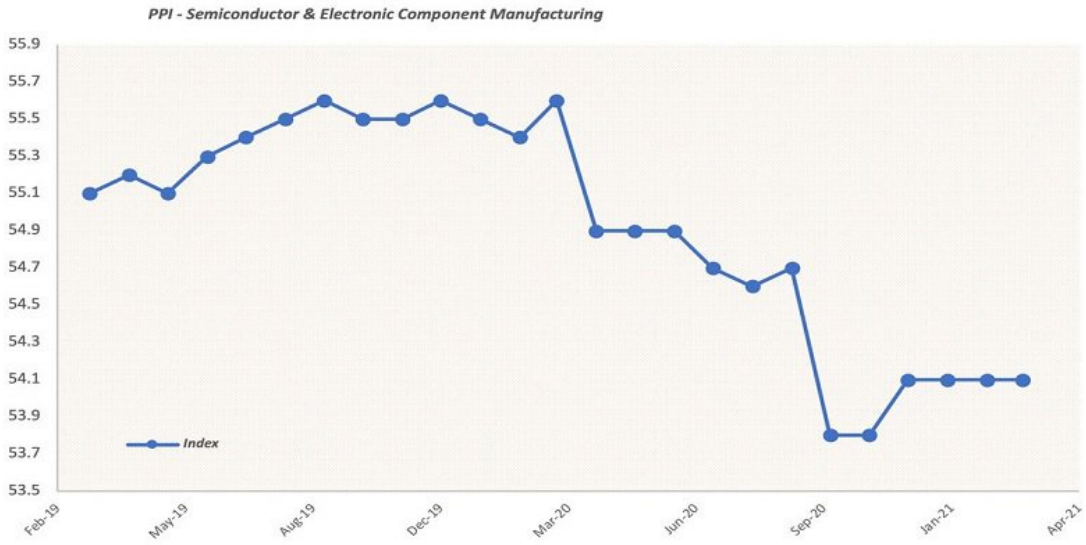


**Figure 32.** *Semiconductor Manufacturing(Supply) 2019-2021*

The lack of semiconductors will affect this project by possible lack of supplies needed for this project. If something goes wrong, or if a

component frys, finding a replacement component is nearly impossible without the addition of another year being added to the project. Some companies have a  setback for orders on semiconductors for a year. Luckily, most of the major components have been acquired and with precaution, this will not be an issue.

This can also create issues with budgeting. Due to the shortage of supplies, semiconductors are sold for almost twice as much as they are worth through websites such as Ebay. Though this enables a user to obtain their device sooner, it will deplete their bank account by a lot more than its original price.

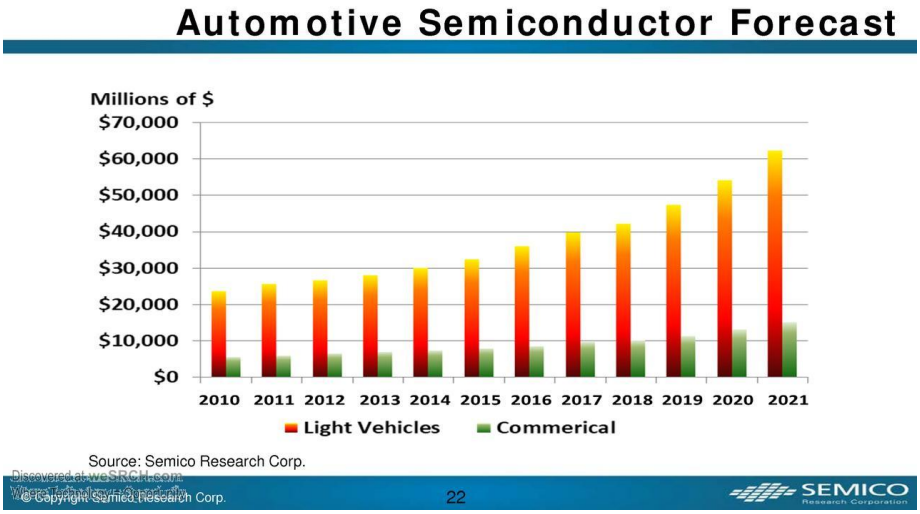


**Figure 33.** *Automotive Demands 2010-2021*

### *4.2.4.2 Narrowing Project Scope*

From the initial concept of HoloMon, the team envisioned a robot that would follow the user around on the floor while projecting a holographic display of a Pokémon (formal IP access requested and acquired) that the user could interact with via gesture based recognition. Additional features for the first iteration of the concept would include a mobile app that would serve as the main back-end communication platform between the camera that would see user's gestures and change the Pokémon to react accordingly as well as an aesthetic feature where the user could change the holographic Pokémon.

However, due to major time and economic constraints, the team decided to cut out the robot and make the hologram stationary. This decision was also influenced by the research done and current holographic technologies are both limited by the amount of light in a room as well as the amount of vibrations present. This made the idea of placing a hologram on a moving platform very challenging given the constraints. So

after narrowing the project scope, HoloMon's initial prototype will be a stationary holographic Pokémon that the user can change via the mobile app and interact with using gesture based recognition.

Based on the information in the previous subsection that thoroughly discusses semiconductor shortages, it was therefore very unlikely that we could pick the best or most ideal components for each part of the project. This in combination with the given time constraints of completing the project in a semester after testing the individual subsystems and doing research in the previous semester, it is probable to assume that HoloMon will be a crude prototype of the original concept.

### *4.2.5 Manufacturing and Sustainability Constraints*

Manufacturing constraints serve as a hinder to the mass production of our product. In the manufacturing process, the main bottleneck will be the method in which the hologram is created. It uses the pepper's ghost technique so creating a self-contained product that can be manufactured will be difficult. In this design, an LCD is used in the form of a television. In manufacturing, a television is overkill for what is needed, however due to budget constraints resources already obtained were used instead of purchasing new components. In mass production, integrating a LCD in this product will be necessary to reduce manufacturing costs and create a self-contained system.

One thing to also keep in mind about this constraint is the quality of the television compared to other devices such as a computer monitor. This television is rather old and has less quality when it comes to images. The TV is also much larger than a computer monitor. With this massive screen being used to project the pepper's ghost image, the team will have to make a large housing unit to encompass the whole perimeter. Further description of what this will look like will be shown in the next chapter.

The sustainability of this product on the market will be highly determined by the cost. While it is not the most technologically advanced product on the market, the cost may make it competitive. When considering sustainability we want the parts selected to be common and are easily obtainable, which we have chosen for our particular product. We also want our product to be durable so as to increase its lifespan, we will take this into consideration in the design.

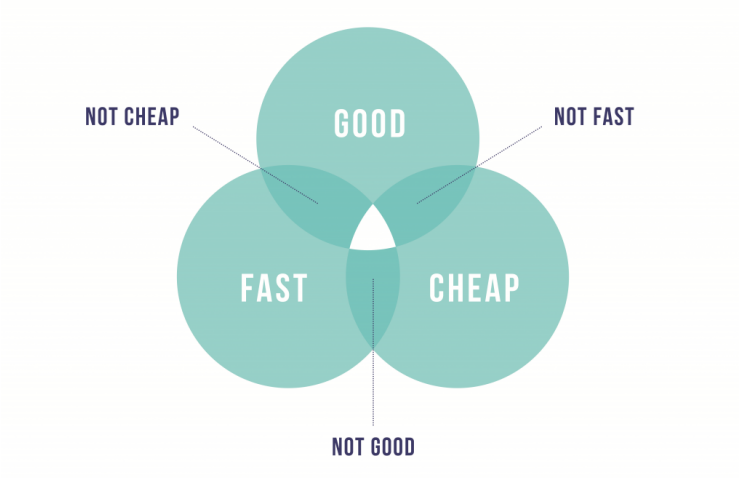


**Figure 34.** *Constraints Examples*

# 5. Project Hardware and Software Design Details

The Holomon will implement a multitude of interconnecting features. These being motion sensing, computer vision, and holographic display. In this section, the implementation of these features are broken down into the design and development stages.

## *5.1 Initial Design Architectures and Related Diagrams*

The concept of HoloMon is pictured in multi-viewed angles in the following figures. The first displays the approximate size of the device relative to an average sized human as well as the front facing aesthetic.

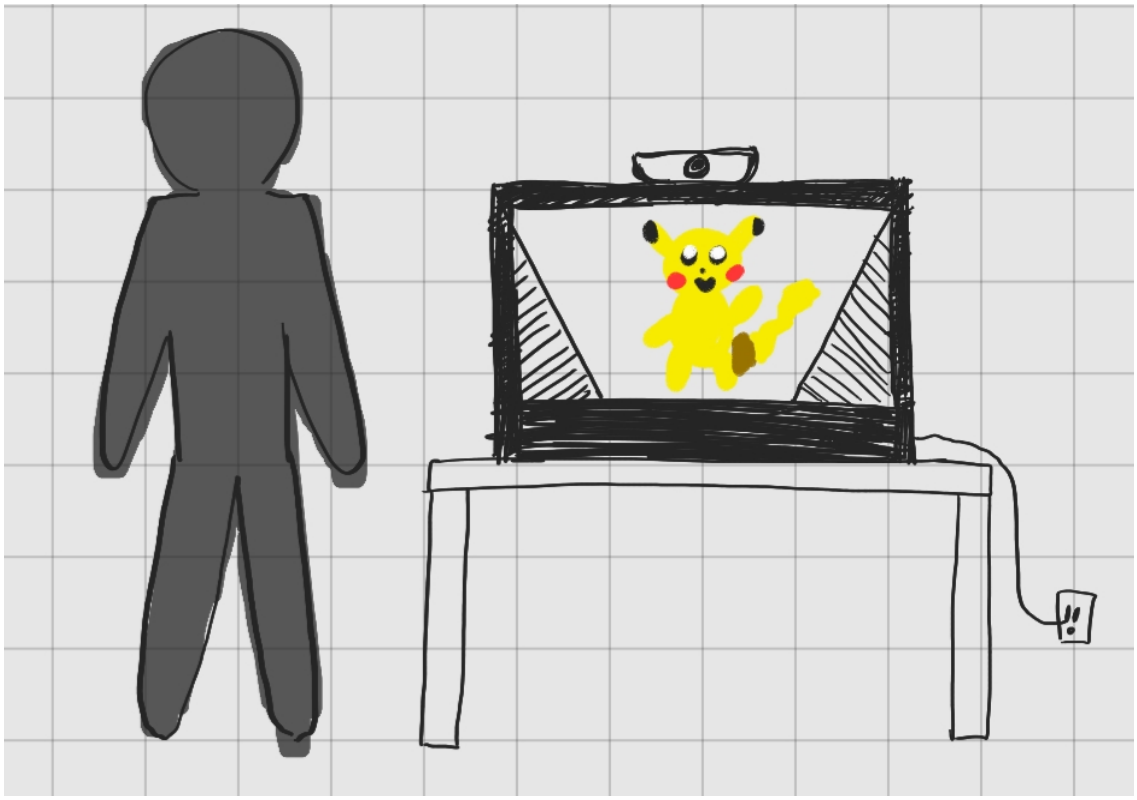

**Figure 35.** *Relative Size Perspective of HoloMon*

The second image shows how the Pokémon would react to a type of hand gesture recognized by the camera that appears above the housing unit of the device. While this is not the exact reaction the Pokémon would take nor paired with the correct gesture, this is simply an example of how waving a hand to the hologram would change the asset and react accordingly.

There will be about 3 - 4 gestures that the user can make to interact with the holographic Pokémon in the HoloMon device. It is likely that we may include other gestures to control the type of Pokémon being displayed in the viewing area for the simplicity of the project.

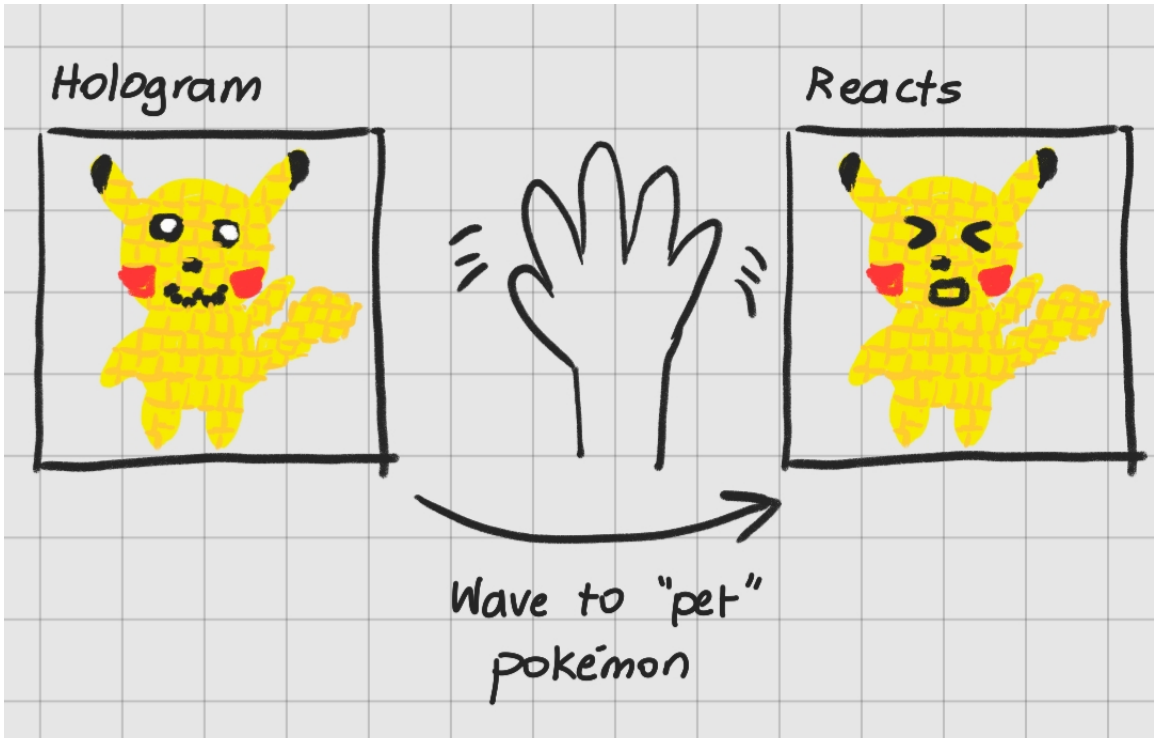


**Figure 36.** *Example of Hand Gesture Interaction*

This third image presents the two-layer view of how the HoloMon device will be created. The outer housing unit, most likely made out of black spray-painted plywood, will be nestled on top of the plexiglass pyramid and TV.

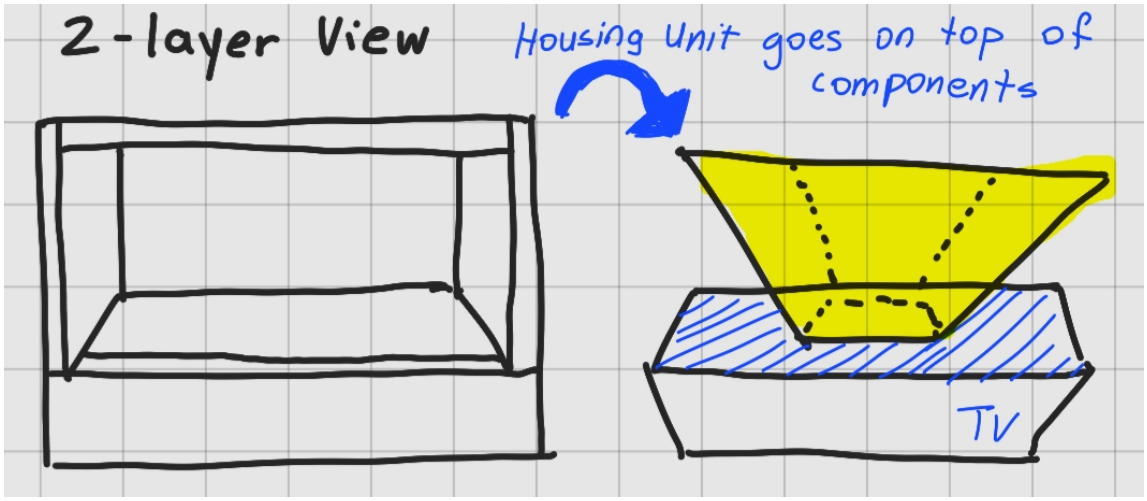


**Figure 37.** *HoloMon Two-Layer View*

This last diagram hosts the three-quarter view and the top-down view of the HoloMon device. These angles were important to showcase so that the evaluator could notice that the housing unit has a removable back panel that hides all the

technical components and wiring to make the device look professional. It also showcases some details such as two of the components using a wall outlet for a power supply while the third uses a 9V battery. The camera is outward facing, so that the viewer when looking at the hologram from the left side can be noticed by the motion sensor and the camera.

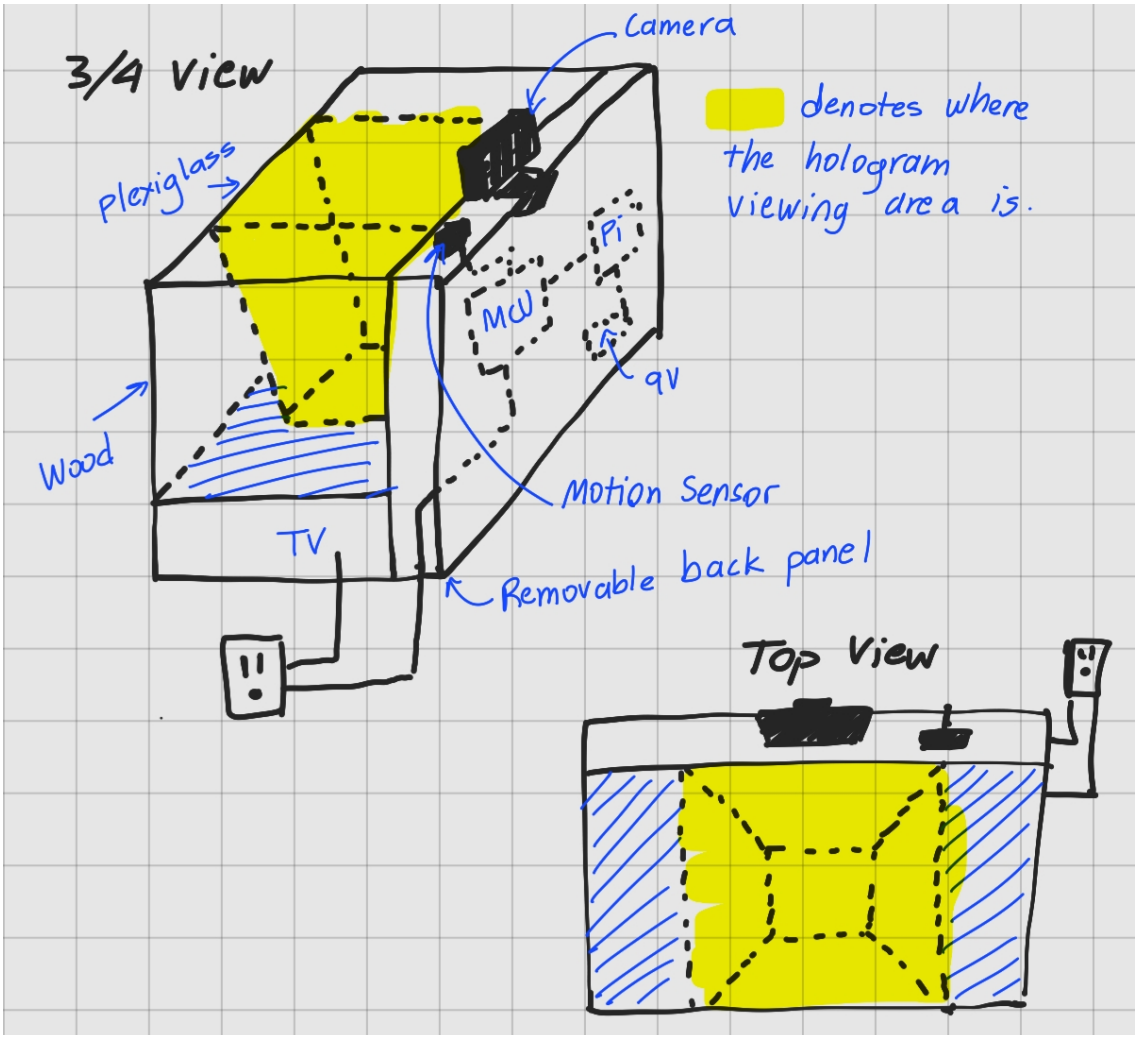


**Figure 38.** *Multiple Perspectives of HoloMon Device*

## *5.2 Software Design*

There are many layers of software within this project and many, if not all, components are dependent on it. The main components that are heavily dependent on software are the Amazon Webservice EC2 instance, client mobile application, raspberry pi (and its modules), and Arduino (and its modules). These components will interact through various network communication protocols as well as direct communication through pins and wires.

### *5.2.1 Cloud Instance*

Amazon provides various free tier services that let their users interact with their services. The most basic service would be Amazon Web Service's Amazon Elastic Computing Cloud surfaces, known as EC2. Amazon Elastic Computing Cloud (Amazon EC2) is a cloud computing service that offers safe, scalable compute power. It's intended to make web-scale cloud computing more accessible to programmers. The easy web service interface of Amazon EC2 allows you to quickly obtain and configure capacity. It gives you complete control over your computing resources and allows you to run on Amazon's tried-and-true computing infrastructure[37].

EC2 will host both the API holding the business logic of the application as well as the database. The only method of communication to the API will be HTTP and the API will connect directly to a database server instantiated within the same EC2 instance. Amazon does provide other methods and services for managing databases but an EC2 instance is sufficient, safe, and scalable enough to hold both systems without any major interruption.

For better privacy of the user data and the business logic of the application, a user group policy can be established to prevent any unwanted users from being able to access the cloud instance. In order to SSH into our servers to either debug or maintain the application, the developer would need to provide the team with both their IP and a generated key. The only reason to SSH to the server would be to pull any changes made to our GitHub repository containing the source codes and/or to alter any of the database tables.

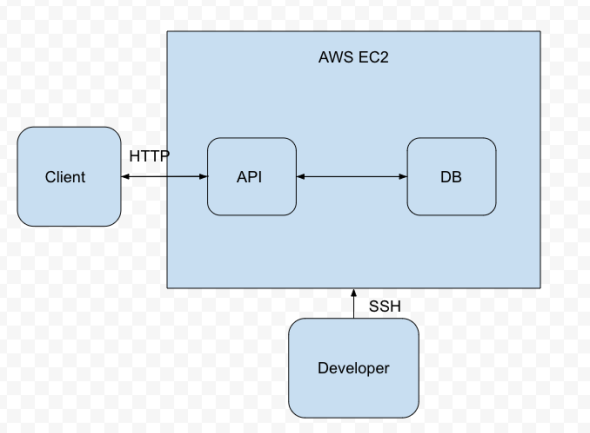


Figure 39. *AWS, Client, Developer connection*

## *5.2.2 Database*

The database would be configured to let any local path access it. This may seem like not the most secure policy but the EC2 instance would not let anyone without the correct credentials or IP tunnel themselves in. This method cuts down on complexity as well as any other bugs that may arise from overcomplicating this procedure.

The data stores would be simple user credentials such as User Id, hashed password, username, email address, date of birth, last login, last Pokémon fetch, first name, and last name. For the assets, a simple Pokémon name with an asset path is sufficient for the API to gather the asset and send it to the client. This would be stored in a document object fashion as MongoDB is not a relational database.

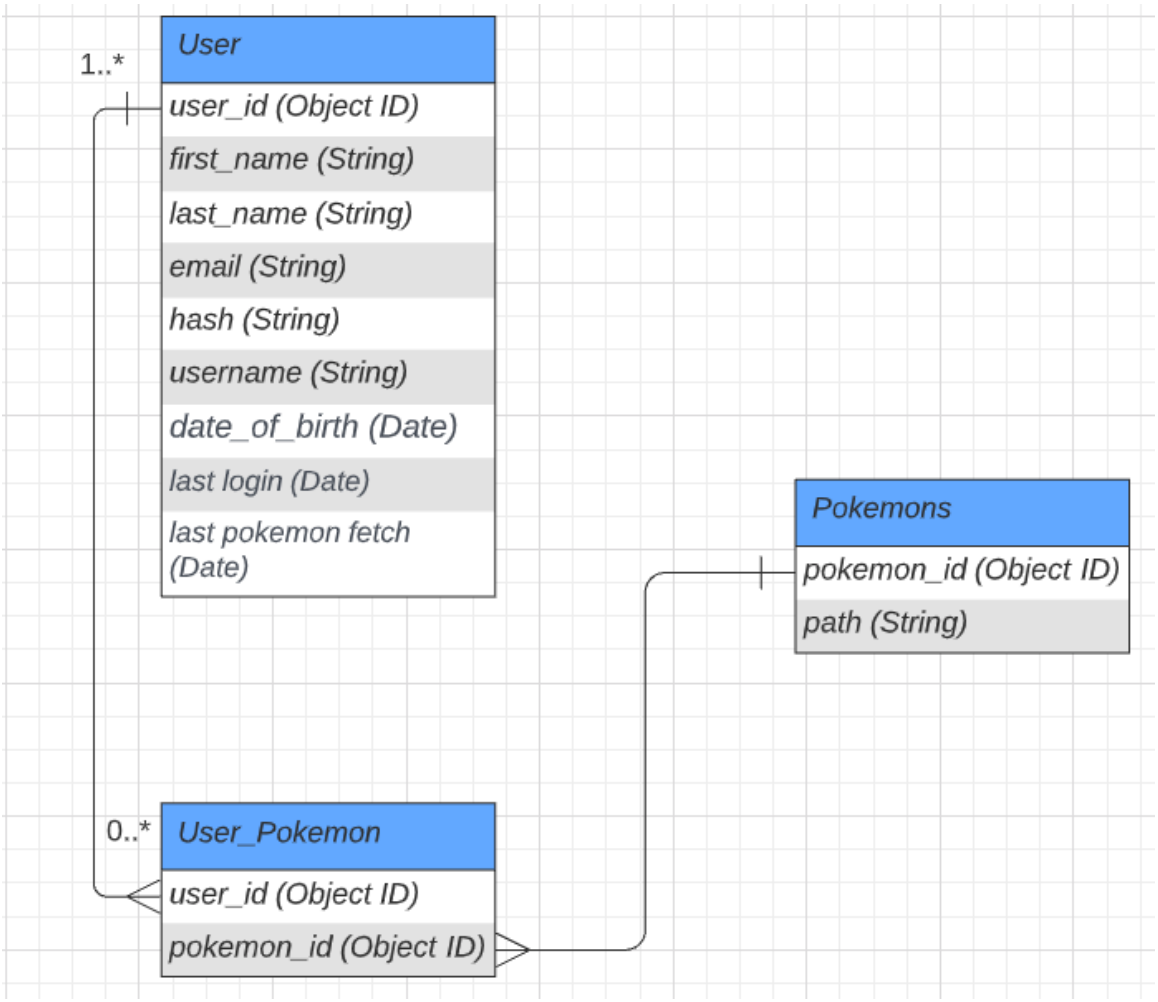

**Figure 40.** *Database Diagram*

Password hashing and salting would be necessary to prevent storing passwords as regular text. This method ensures that passwords would be safe even in the case of a data breach. Salting is a term used to describe the process of hashing passwords. It's a one-of-a-kind value that can be appended to the end of a password to get a different hash value. This adds an extra layer of security to the hashing process, preventing brute force assaults. A brute force assault is when a computer or botnet tries every conceivable letter and number combination until the password is discovered [38].

### *5.2.3 API*

The application program interface (API) will ensure the connection between the user and the database. The main design pattern between the user and our data is known as a model, view, Controller (MVC) pattern. This is a well-known design pattern mostly used for creating modern web development applications that separate the concerns of the client source code, business logic, and data. (MVC) is a software design pattern[1] that divides linked program logic into three interrelated pieces and is often used for building user interfaces. This is done to distinguish internal information representations from how information is presented to and accepted by users[39].

The API needs to have a series of function calls that enable the creation of an account, secure login, updating credentials, and loading Pokémon assets. Login-in can be simple with the use of Json Web Tokens to authenticate users and authorize them to continue using the API. JWT is a proposed Internet standard for encoding data with optional signatures and/or encryption, with the payload including JSON that asserts a set of claims. A private secret or a public/private key is used to sign the tokens [40].

#### *Authentication*

These tokens can be generated using python's "python-jose" library which is supported and encouraged to be used by FastAPI. When a user hits the /login endpoint and the login is successful, the API returns a JWT with the date of creation metadata. This JWT is then used for every API call going forward to authenticate the user before every transaction. This method ensures that the API can only be used by users who are authenticated and therefore less likely to be exploited. JWT can also have expiration data that ensures that the user is not forever logged in to the application.

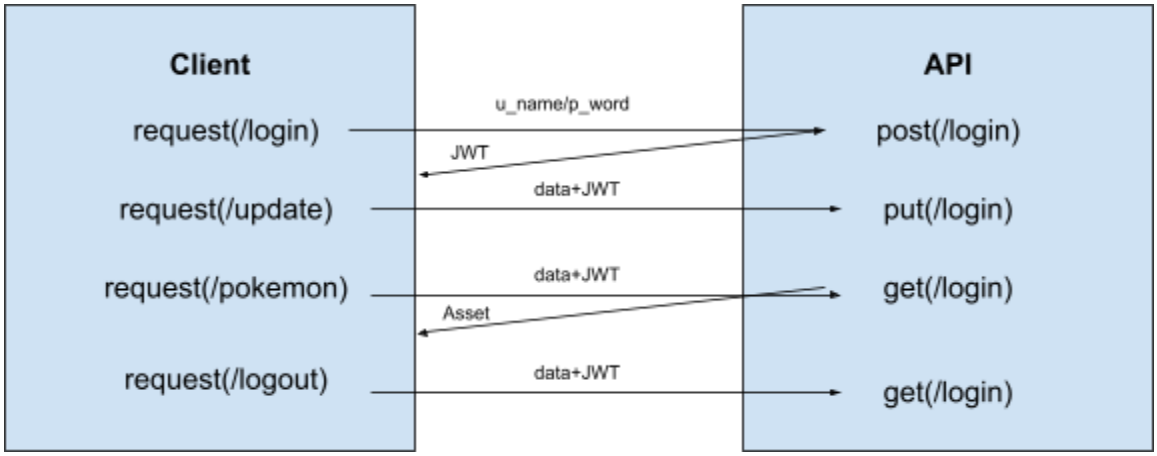


**Figure 41.** *Client, API data flow*

Since security and user privacy are extremely important, Passwords are hashed by the API before being placed in the database. Python's "passlib" is also recommended by FastAPI since it gives the API hashing capabilities with the "Bcryct" algorithm. Said library also grants the API authentication capabilities without the need to create our hashing algorithm and is generally safer than simple encryption algorithms created by non-professionals.

### *Database Connection*

Connection to the database is fairly simple with a connection string. Since the database is already opened to all applications within the EC2 instance, all the API needs are the correct credentials to be stored in the connection string to create the connection. A connection string is a string in computing that specifies information about a data source and how to connect to it. To establish a connection, it is supplied in code to an underlying driver or provider. While a database connection is most usual, the data source could instead be a spreadsheet or text file. The connection string may contain information such as the driver's name, the server's name, and the database's name, as well as security information like the username and password [41].

## *5.2.4 UI*

The main job of the client application is to provide the user with an interface that they can use to interact with our internal systems. There should be a total of 4 views or pages for this application, the login page, create user account page, update profile page, and the choose HoloMon page.

The login page will simply let the users authenticate themselves. Once the login button is hit, the user client will send an HTTP Post request and wait for a status code. If the status code is not 200, then the authentication failed, else, the

authentication succeeded and a JWT token would be passed back from the API and stored in the client's device.

The JWT token can be either stored using React Native's AsyncStorage which behaves just as a browser's local storage. AsyncStorage is a global key-value storage system that is unencrypted, asynchronous, and persistent. On iOS, native code backs AsyncStorage, which stores tiny data in a serialized dictionary and bigger values in separate files. AsyncStorage on Android will utilize either RocksDB or SQLite depending on what is available [42]. Though the official package backed by Meta might be deprecated, there are many community-backed packages that go by the same name. That JWT will then be used for every API call going forward for privacy and API safety reasons.

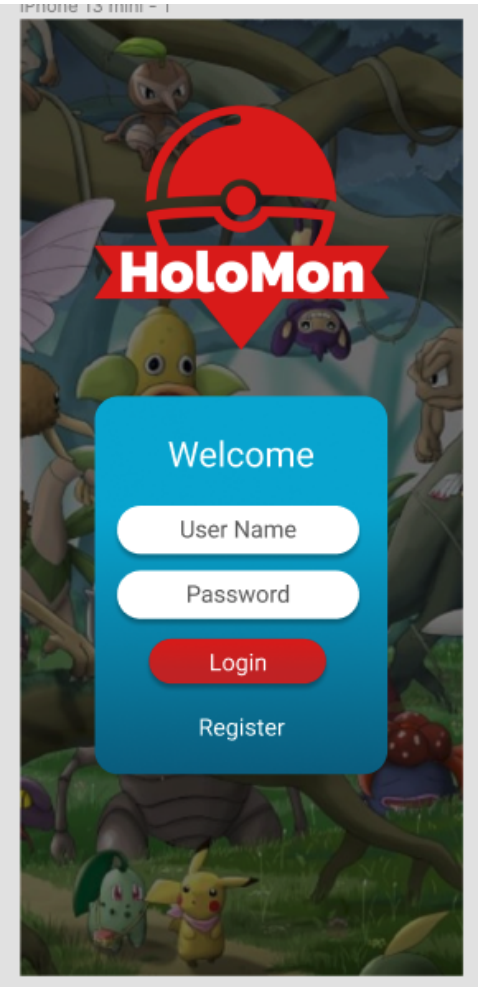


**Figure 42.** *Login Page*

Once the user is authenticated, they will immediately be presented with their current Pokémon and arrows to select other available assets. These assets will then be loaded to the Raspberry Pi and eventually be displayed with the hologram technology.

This page requests the API for the current Pokémon the user can load, the assets will then be displayed accordingly. This API call also used the JWT for authentication.

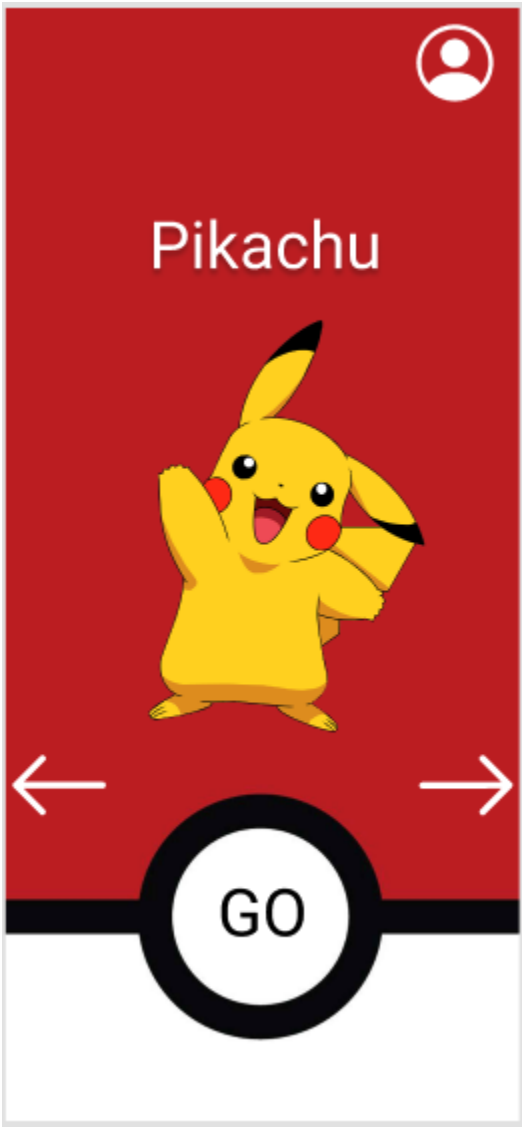


**Figure 43.** *Pokémon Selection Page*

The edit user profile page lets users update their credentials including their email, in case there were any typos on creating the user or to better represent themselves. Once the update button is hit, the fields in the form will be transferred as JSON through a put request with the JWT stored in the user's device. If the status code is 200, then the data is correctly updated and the user should be notified. Any other status code aside from 200 should warn the user of a failure in the update process.


**Figure 44.** *Update User Page*

The create account page is similar to the edit account page but the password is added to the form. The password is checked in the client for certain criteria and then checked again once within the API. This gives the user a better and more responsive experience when inputting the wrong password in the “check password” field without the need to hit the API and wait a couple of seconds for the response. Checking in the API gives this feature an extra layer of security as someone trying to send malicious data to this endpoint would be stopped.

This Put Request does not need a JWT since the user is yet to create an account, therefore, is not authenticated.

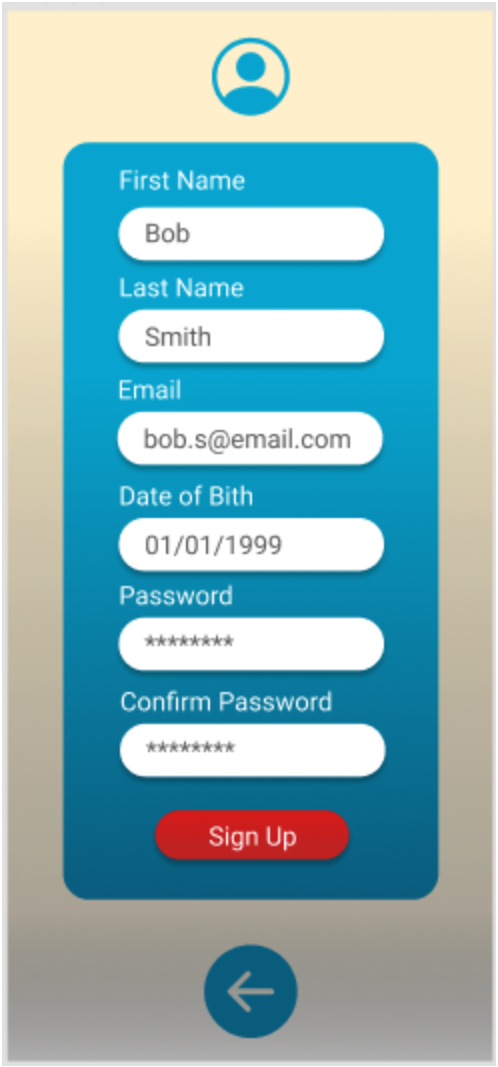


**Figure 45.** *Create User Page*

## *5.2.5 Camera to Server Connection*

The biggest software obstacle is the connection between the Raspberry Pi and the Cloud Server. There are a variety of ways to go about this but the most efficient and easy to implement way is by establishing a connection between the Mobile Client Application and the Raspberry Pi through Bluetooth, for authentication purposes, to then enable the Raspberry Pi to directly connect to our server through HTTP.

The mobile client would authenticate the user through HTTP first then try to establish a connection to the microprocessor and send the user's credential token. Once the token is inside its storage, the microprocessor would then make an HTTP call that downloads all the user's assets from our cloud server which will then be displayed to the user using our technology.

There should be no need for any more HTTP calls from the Raspberry Pi to the server as the data is static from that point on. The only other call expected from the mobile client to the microprocessor would be the logout flag that will direct the Raspberry Pi to delete the user's assets and their authentication token.

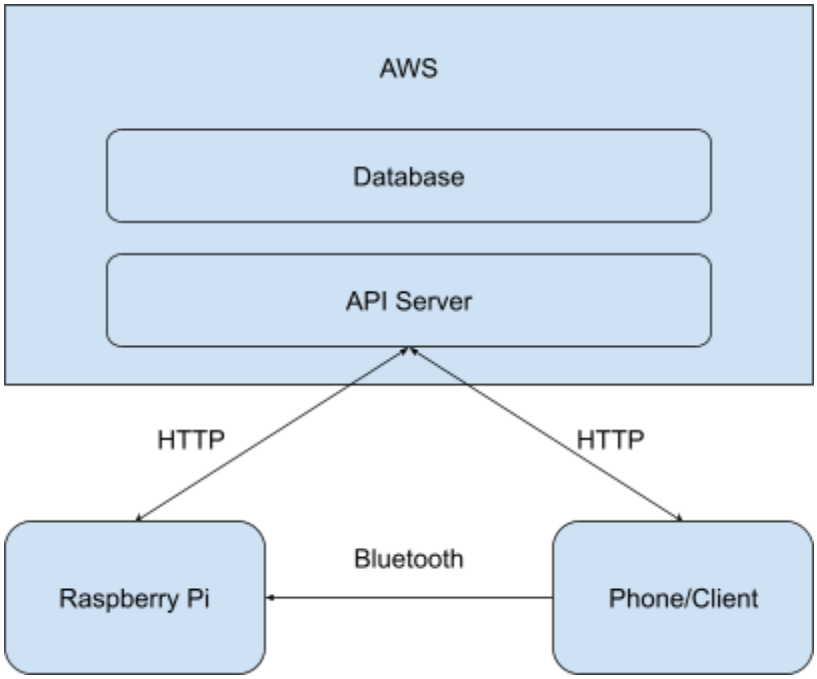


**Figure 46.** *Software Connection*

The Bluetooth connection can be set by using the serial library that can be downloaded to the Raspberry Pi and using the react-native-ble-manager library for React Native, installed by any javascript packet manager.

### *5.2.5 Microcontroller*

The software for the system can be coded in Arduino IDE. The open-source software makes coding and uploading to our board very easy. A USB 2.0 port will be used to connect the computer to the board to configure the firmware and upload it. There is only one sensor connected to the board and this is the only data that the microcontroller needs to read. The microcontroller will then feed the input received from the sensor to the Raspberry Pi for processing.

There are two parts to the code, the first is configuring the pins and the second is running the code that will be executed. There are three pins that need to be configured, them being VCC, ground, and output. In the loop, digitalRead will be used to read the value of the pin that is connected to the output of the sensor. The value will either be high or low (high if the sensor detects motion). If the value is high then Serial.println will be used to output "Motion detected" to the terminal, for testing purposes.

## *5.3 First Subsystem: Gesture Recognition with Camera*

The first subsystem will be responsible for detecting hand gestures using the USB webcam and machine learning. This subsystem is important for the main portion of the project because it will allow users to interact with the HoloMon. The components used in this subsystem are the Raspberry Pi 4 Model B and the Logitech HD Pro C720 Webcam.

## *5.4 Second Subsystem: Motion Sensing and Camera Control*

The second subsystem will be responsible for sensing when a user appears in front of the system and turning on the camera when it senses it. This will allow the camera to not be continuously on, even when not in use, in turn the computer vision system would not have to be running unnecessarily. The result is a more power and energy efficient design.

The main components responsible for this subsystem are the Arduino Uno R3 microcontroller and the motion sensor.

## *5.5 Third Subsystem: Raspberry Pi to Display Control*

The third subsystem will be responsible for ensuring that the Raspberry Pi can communicate between the display control and the mobile application. This will be the primary way that the backend handles the requests from a user changing Pokémon in the app to having the hologram change accordingly. Additionally, it will be important for the backend to handle downloading all the assets onto the Raspberry Pi to be displayed on the TV.

The components for this subsystem includes the Raspberry Pi 4 Model B, the TV with HDMI capabilities, the HDMI to micro-HDMI cable, and the wall plug to USB-C power supply for the Raspberry Pi.

Since the project is stationary, it will not be necessary for the Raspberry Pi to share the 9V battery that powers the Arduino and instead can be plugged into the same wall plug outlet that the TV will need. The TV needs a wall plug since it draws a lot of power to continuously display images and would not be feasible otherwise.

## 5.6 Fourth Subsystem: Raspberry Pi to Arduino Interface

Since the motion sensor will be on the Arduino/PCB and the camera on the Raspberry Pi, there must be a USB connection between the Arduino and Raspberry Pi so these two interfaces can communicate. Ther Raspberry Pi must apply some kind of code referencing to obtain information from the Arduino. The Arduino will hold the sensor and have a code that is detecting whether this sensor is detecting something. Once detecting the sensor and collecting the data, the Arduino sends information to the Raspberry Pi. If the sensor picks up motion, it will quickly turn on the camera to interact with the user. The arduino must be coded to pick up the data from the sensor and respond back to the Pi. So the arduino will notify the pi once data is collected from the sensor, and the Pi will react by turning off the camera.

## 5.7 Summary of Design

This section provides the final summary of the design for the HoloMon project.

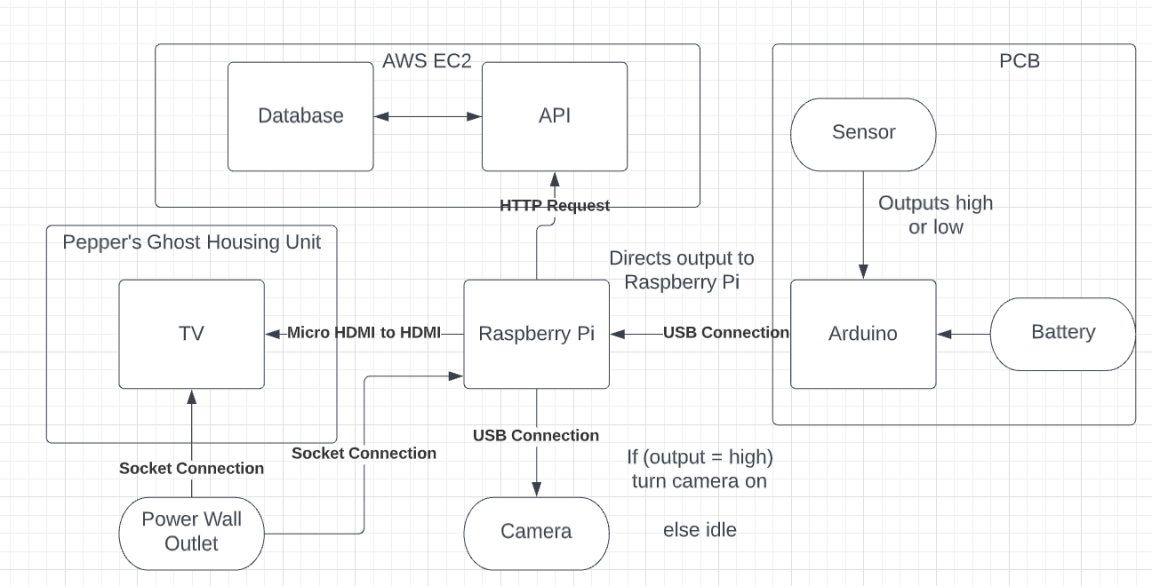


**Figure 47. Summary of Design**

At the heart of the PCB is the Arduino microcontroller, it manages the motion sensor. It is powered by the battery on the PCB. The microcontroller has a direct USB connection to the Raspberry Pi which performs most of the controller logic in the system. Through this connection, the output of the sensor is received by the Pi and if that value is high the camera is turned on. There is a Micro HDMI to HDMI connection between the Raspberry Pi and the TV, which is the chosen display to achieve the Pepper's Ghost Illusion. Both the TV and Raspberry Pi are powered by the wall outlet. Upon request from the mobile user to change the displayed Pokémon, the Raspberry Pi performs an HTTP request to the API, retrieving the data from the database.

# 6. Testing and Integration

Chapter 6, Testing and Integration, focuses on primarily creating the PCB schematic and how all the subsystems (see Chapter 5) will integrate with each other to make the final product. Here, the main subsections detail the PCB Prototype, the PCB Vendor or Distributor, and the Final Coding Plan. Furthermore, details of how each is accomplished is within each subsection.

While this chapter mainly discusses the early stages of the testing, some methods to be used in future stages will be explained between chapters 6 and 7.

## *6.1 PCB Prototype*

This section will display the PCB board and placement of all the components on the board. The PCB will mainly be composed of Arduino Uno components containing: a microcontroller, capacitors, a battery holder, a battery, ground, an LED diode, a voltage regulator, a crystal, and resistors. The type of components utilized in the PCB will be listed and described below. Also connections to each component will also be listed and any necessary explanations for connections will be explained. The schematic will be created in EAGLE as well as the BRD scehmatic. This is a rather simple PCB board and will have no complex specifics to its design.

When creating connections on the PCB, there are three main divisions of the schematic.

- Power
- MCU
- Jumper Connections

These divisions are reflected in the following subsections 6.1.1, 6.1.2, and 6.1.3.

### *6.1.1 Power*

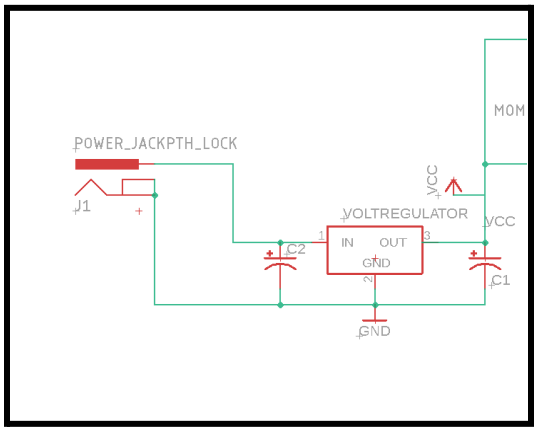

**Figure 48.** *Power Schematic of PCB (Left side)*

This section of the design will power up the entire PCB board. On the PCB board there is a power jack that will connect to a 5V Voltage Regulator that will take from a 9V power supply and regulate it to 5V for the Adrduino. The capacitors on the side of the arduino are assisting the voltage regulator storing energy for it to be utilized later. The voltage will come from a battery which will be connected to a Power Jack and regulated to the VCC. The VCC powers up the entire PCB.

## *6.1.2 Jumper Connections*

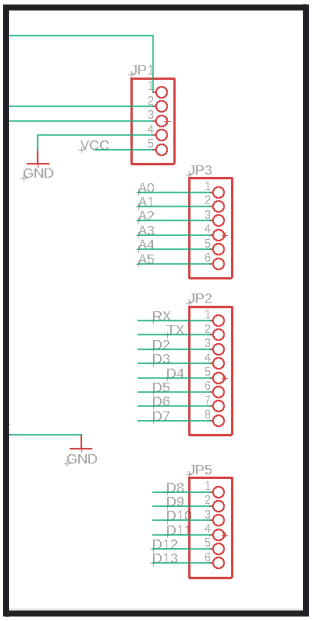


**Figure 49.** *Jumper Cable/Connection*

Since the PCB must connect to the Raspberry Pi, there must be some form of USB serial connection for these two to connect. One useful component for this is the CP2102 USB to TTL adapter. The CP2102 will be connected to the first jumper seen at the top. The three jumpers below the first are all connected to the MCU accordingly for any other necessary connections used for the project. Jumper two will most likely be utilized for the motion sensor for the power saving mode. Most likely D2 will be the pin to connect the motion sensor.

## *6.1.3 MCU*

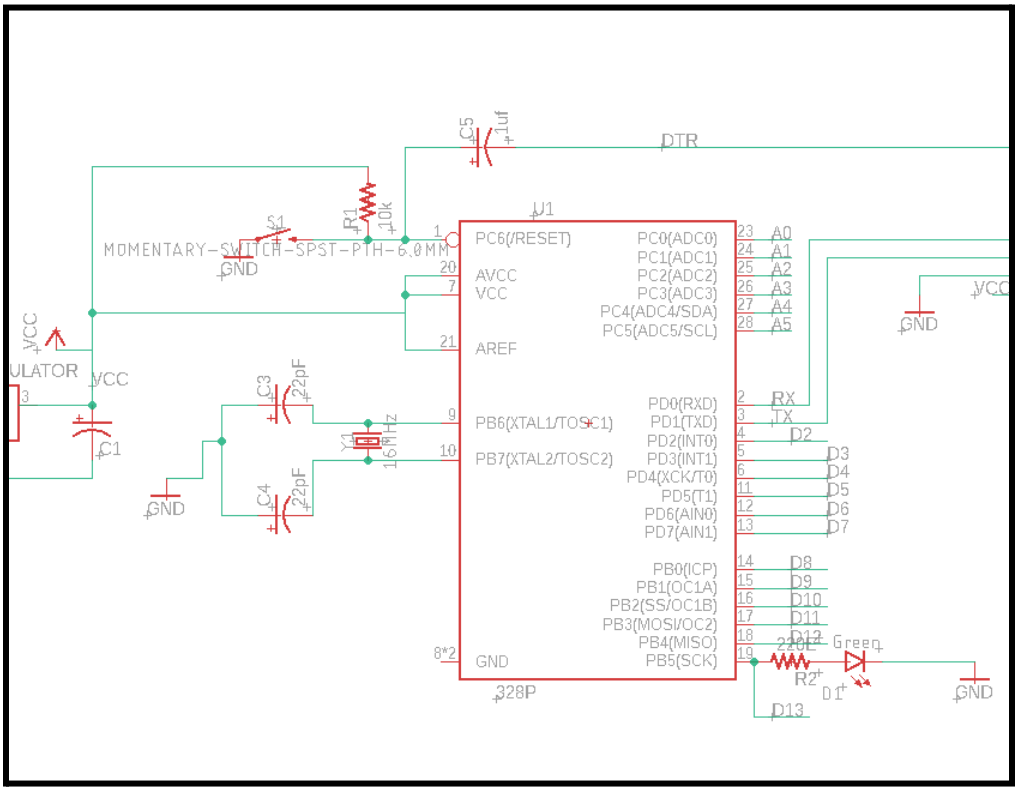


**Figure 50.** *MCU Schematic*

This MCU is nothing out of the ordinary especially since it is Arduino based. The MCU selected is the ATMEGA328 with 28 pins. One important feature for the MCU is the crystal which has been set to 16 MHZ which connects to capacitors of 22pF that connect to ground. Crystal oscillators act as a clock source for the MCU, a very vital component for the MCU. As stated before the pins A0-A5 will be connected to the first jumper set which will most likely hold the USB. This USB will be connected to the Raspberry Pi. There is a switch connected to the VCC and resistors and capacitors placed in necessary areas for the PCB to work correctly. One thing to note is another connection that has not been discussed yet. The HC-SR501 is a vital connection for the PCB and its sole purpose. With the HC-SR501 sensor, the sensor will communicate through the arduino and the Raspberry pi will take the data from the arduino. Once it does this, it will signify the camera to turn on if there is activity and stay off when there is not. This motion sensor is crucial for battery saving of the project. How the HC is connected to the PCB is through the 5V and GND and then to D2 which is seen above. This is a rather simple connection for the PCB and may be able to be soldered on.

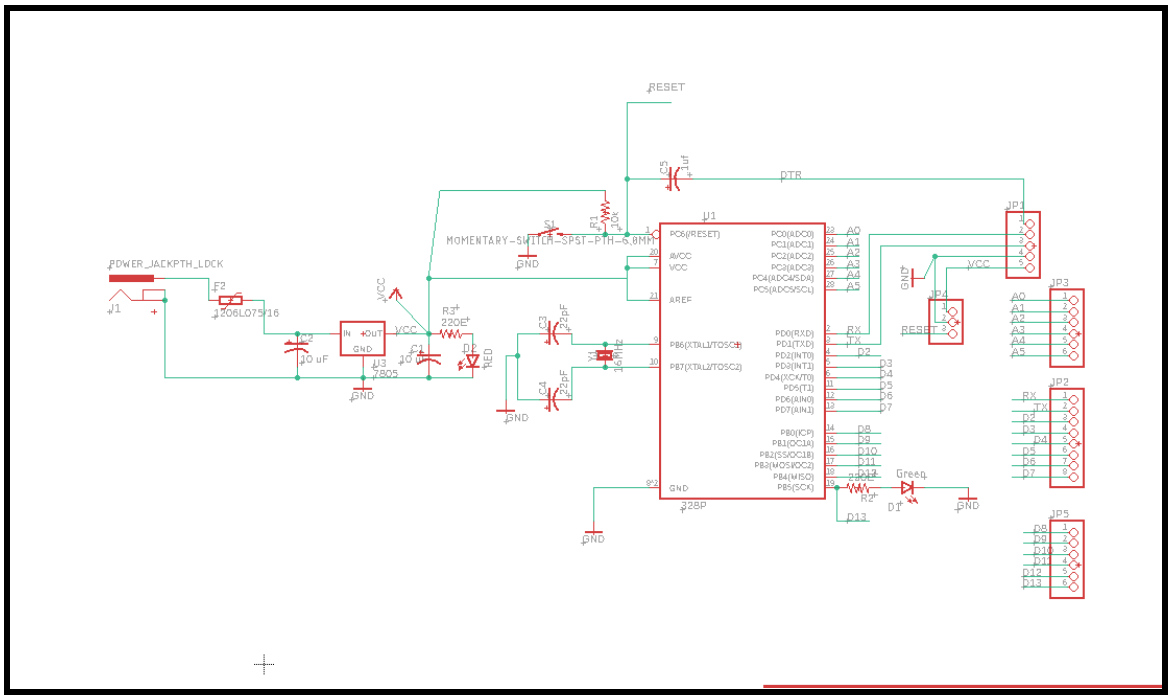


**Figure 51.** *Entire Schematic*

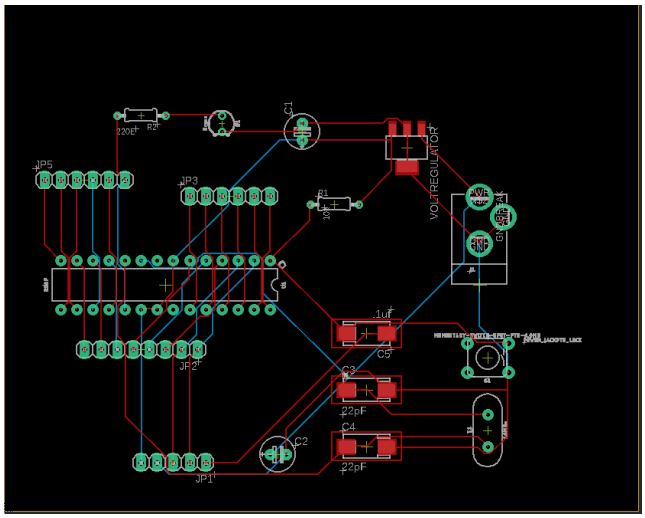


**Figure 52.** *BRD Schematic*

The following shown above is a representation of the Arduino laid neatly on a PCB board. Once all the connections are routed and all the correct parts selected, the PCB will be ready to be manufactured. Placement will most likely be reoriented later but for now will take this placement. For now, it is easy to see that the MCU chip is on the middle left surrounded by the pin configuration. The three capacitors are close to the voltage regulator and the power jack. The diodes and resistors are at the top of the frame. All connections of the PCB are shown in the image above whereas the schematics may not have shown these connections as clearly.

## *6.2 PCB Vendor/ Distributor*

The next important step for having a PCB as part of any project is finding the right company to manufacture the board. This is because many companies might have a minimum required quota to purchase while others may be based in other parts of the world with slow turnaround time for the pace of this project. Also an important factor for this is the project's budget, although usually printing PCBs are not expensive, it can quickly add up if insufficient testing of schematics prior to manufacturing are left unresolved.

### *6.2.1 Saturn PCB Design*

One of the vendors the team is researching for this project is the Saturn PCB Design Center located in Orlando. Saturn PCB Design provides electronic engineering and manufacturing. The company delivers PCB straight to doorsteps once the engineer has sent in the PCB. This company does a lot from PCB services to software development. They offer many different materials for PCB and have designs that can be used for space, medical, and or military.This was the best rated vendor and also the closest. Perks of selecting a close vendor includes being able to stop by the facility if there are any complications with the PCB. Another obvious reason for considering this Vendor is that it will most likely not take too long to distribute to the team once it has been ordered. This is ideal since once the project starts, it will be best to have the PCB to help start the building. This is a great distributor of PCB with great ratings and the PCB will be obtained from this location.

### *6.2.2 JLCPCB*

This company is worth looking into, since they sell PCBs for cheap, giving multiple copies of the same PCB. Additionally, they are growing quickly to become the leading standard for PCB manufacturing and delivery since being founded in 2006. Large entities such as IEEE create partnerships with JLCPCB and have ongoing discount rates for members in their organization when purchasing from them. They are known to have reliable and easygoing customer service while having high quality prints for sustainably low prices. The only downside is that this company is based in China and thus the shipping rate is extraordinarily high.

# 7. Project Prototype Testing Plan

This chapter, project prototype testing plan, is to ensure that the groundwork is created for combining multiple subsystems into one complex and functional project. The main sections that follow are the hardware test environment and the hardware specific testing discussed in greater detail.

## *7.1 Hardware Test Environment*

For this subsection, hardware test environment, it outlines the environments in which individual team members integrated their components together. The team used whatever locations were available to them. In many cases, since the subsystems did not require special ventilation or other restrictions to test, the team used their respective living quarters to test different communications between devices.

## *7.2 Hardware Specific Testing*

In this subsection, the hardware specific testing is covered to ensure that the individual subsystems are working as intended. Also they are covered to provide the necessary knowledge to perform the next semester's objectives of creating more complex integrations between them. The tested subsystems are as follows: the connection between the USB webcam and the Raspberry Pi, the Arduino and the motion sensor, the Raspberry Pi and the TV, and lastly the Raspberry Pi and the Arduino. The PCB itself is not included in testing, since the manufacturing of the board will be left for the next semester in Senior Design 2.

### *.7.2.1 First Subsystem Testing*

The gesture recognition with camera subsystem needs a Raspberry Pi 4 (Model B) to connect. The first step to testing this subsystem was plugging in a microSD card into a computer via a card reader and downloading a Raspberry Pi OS (the recommended is a port of Debian Bullseye with the Raspberry Pi Desktop) to the card. After this step was concluded, the second step is to eject the microSD card and plug it into the Raspberry Pi board. The third step is to connect the Raspberry Pi board to a mouse, keyboard, monitor, and using the given power supply, go into the Pi OS BIOS looking software to enable SSH. Since 2016, they have as default disabled ssh for all manufactured Raspberry Pis to prevent hackers from easily using other people's Pis as internet bots.

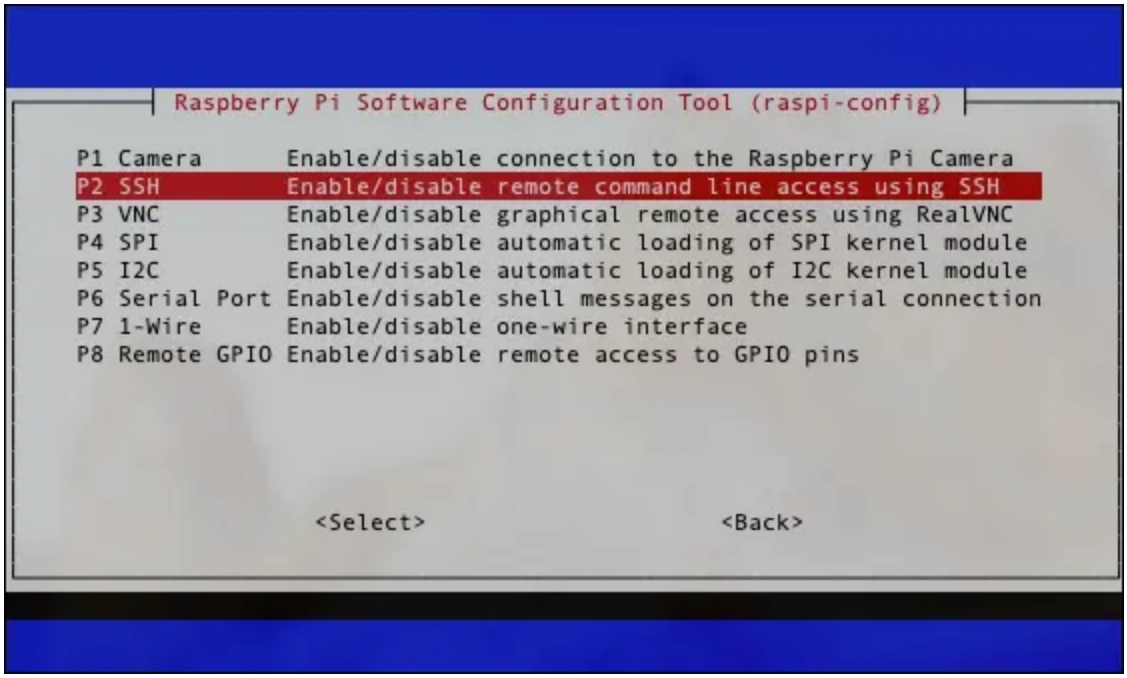


**Figure 53.** *SSH on the Raspberry Pi Configuration Tool*

After enabling the SSH directly on the Raspberry Pi, the next step was to verify that the default hostname was kept for the board as “raspberrypi” using some commands in the terminal. Once those were configured properly, the next step was to unplug the peripheral devices and reconnect the Pi to a computer. There are a multiple number of ways to SSH into something, but for this instance, PuTTY, a popular choice that is a “free and open-source terminal emulator, serial console and network file transfer application” was used to SSH [50]. After creating my own username and password to SSH into the Pi, the next step was to see if the Pi could recognize the plugged in USB webcam. Alternatively, it is possible to SSH into the Pi using other terminals such as Windows Subsystem for Linux, Windows PowerShell, and even the Command Prompt.

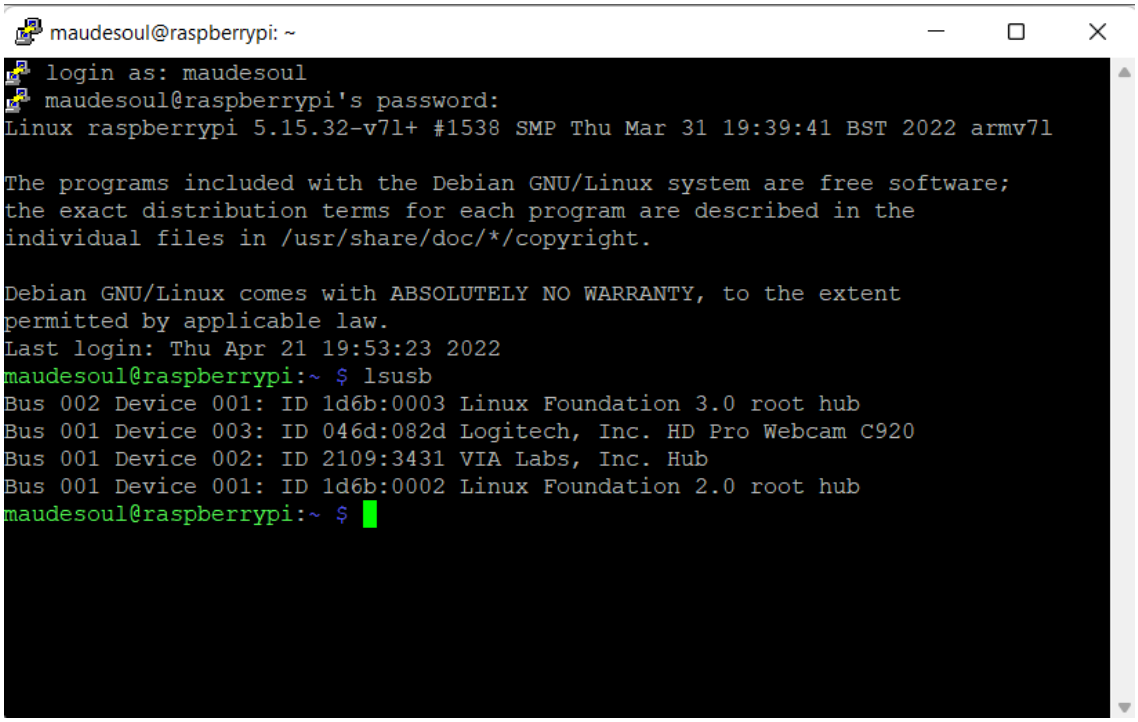


**Figure 54.** *Successful SSH into the Raspberry Pi and Finding Webcam*

After the Pi recognized the webcam was plugged in, the main testing objective for this subsystem was to see if the webcam could take a picture via commands from the Pi. This initial prototype testing will set the foundation for utilizing the webcam to capture gesture recognition from video input. The below picture is a sample image taken using the USB webcam controlled from the Raspberry Pi utilizing “fswebcam” libraries at 1280x720 resolution despite the webcam having 1920x1080 resolution capabilities. Since this is an example image, it was not necessary to have the picture taken at the full HD resolution to conserve on flash memory for the Raspberry Pi.

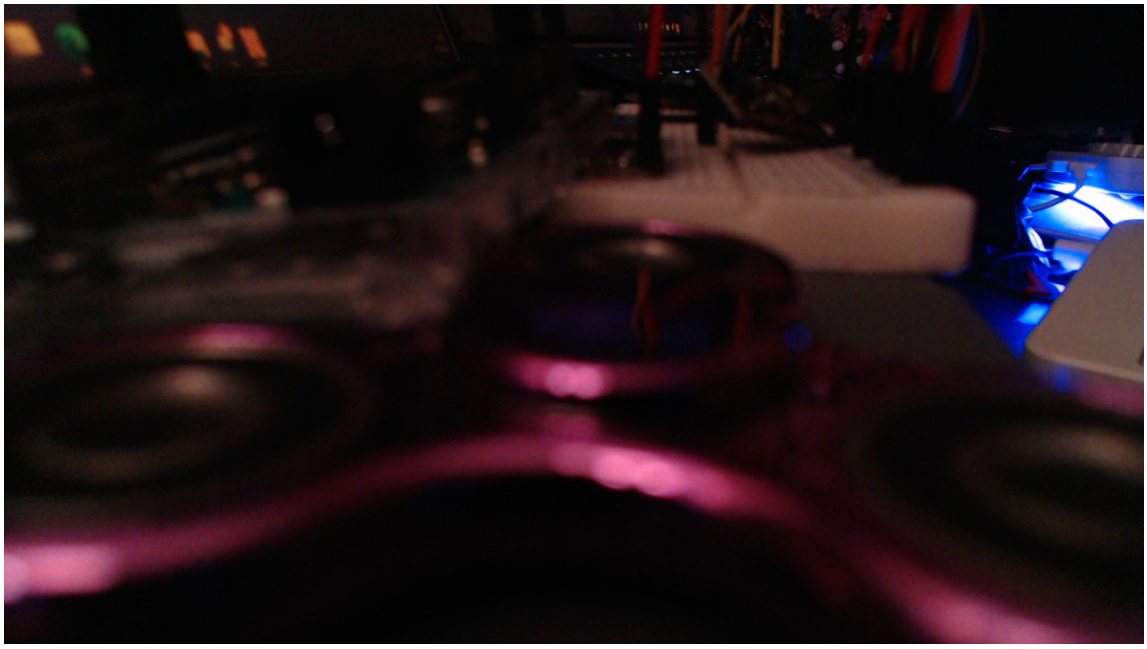

**Figure 55.** *Sample Image on USB Webcam from Raspberry Pi*

## *7.2.2 Second Subsystem Testing*

The second subsystem is composed of the microcontroller and PIR motion sensor. This unit is responsible for detecting the presence of an incoming user and signaling the Raspberry Pi to turn on the camera, beginning the gesture recognition algorithm.

The code will be written in Arduino IDE and uploaded to the microcontroller for breadboard testing.

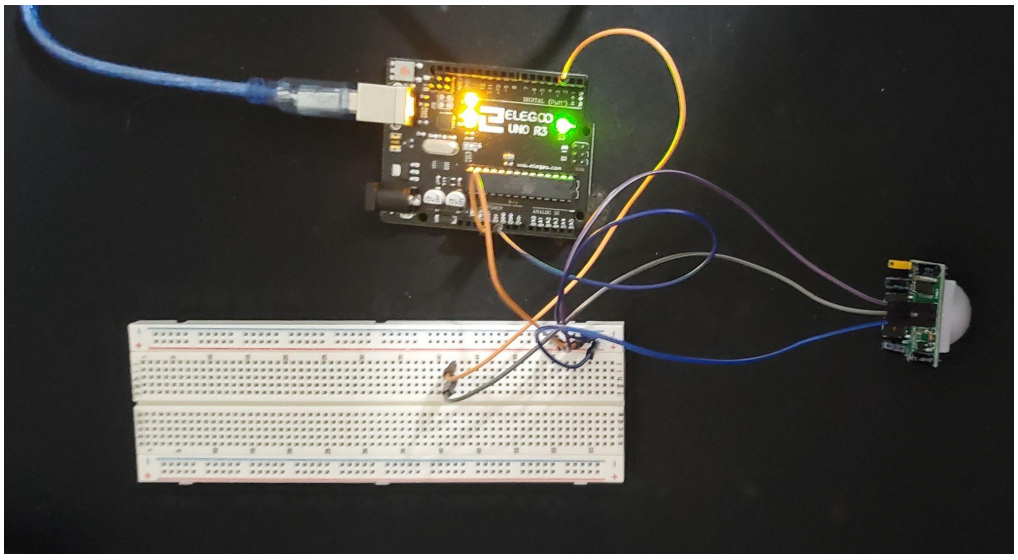

**Figure 56.** *Breadboard test of MCU and Motion Sensor*

The HC-501 PIR motion sensor has 3 pins, VCC, Output, and Ground. In the figure above VCC is connected to the 5V of the Arduino, output is connected to pin 2, and grounds are connected.


**Figure 57.** *Terminal Output of Motion Sensor*

As seen in the terminal output above the subsystem correctly identifies when motion has taken place in its radius. The current range is 3 meters, and can be extended up to 7 meters by adjusting the potentiometer on the sensor. The time delay can also be adjusted by a potentiometer. Time delay will determine how long an output remains after reading it. For this example, the default 2.5 seconds was used meaning 2.5 seconds after motion is detected the camera will remain on. In implementation of the product a higher value can be set and adjusted appropiately.

### *7.2.3 Third Subsystem Testing*

The third subsystem is the connection between the Raspberry Pi and the TV. Since the Raspberry Pi is a single board computer, it has a built in OS that functions like any other computer. This allows the TV to act as a monitor and makes it very easy to set up GIFs and other assets for the HoloMon in the next semester. For this particular setup to work, the Raspberry Pi was powered via the given USB-C to wall plug cable and the HDMI to micro-HDMI cable. The Pi was controlled using a keyboard and mouse via USB-A.

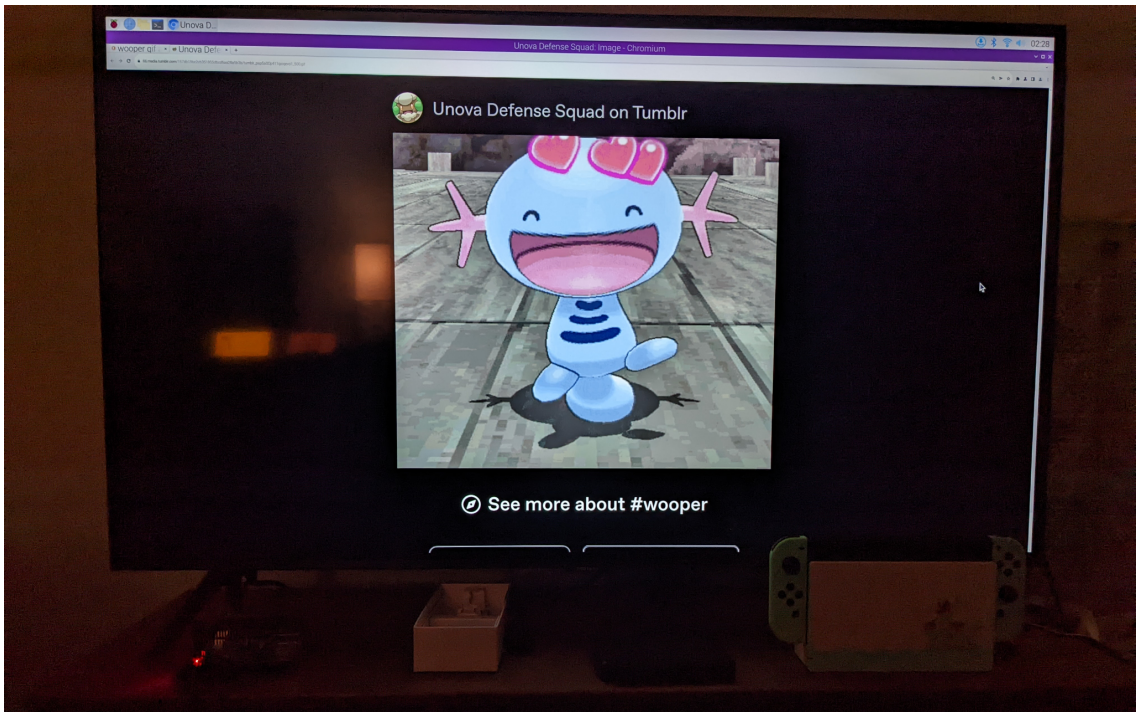


**Figure 58.** *Display assets on TV via Raspberry Pi Connection*

The actual assets for the HoloMon will be stored using the mobile app and will communicate to the Pi using Bluetooth and WiFi capabilities. The software subsystem that controls this will be tested once the skeleton for the mobile application is developed.

### *7.2.4 Fourth Subsystem Testing*

#### *7.2.4.1 One-way Communication Testing*

Testing of this subsystem will only require three items listed below.

-Arduino Uno Rev 3

- Raspberry Pi Rev 4

-USB Connector

The following software apps used for this testing of the subsystem can be seen below.

-Arduino IDE

-NodeRed

As stated previously, the Pi will be programmed to pull information from the Arduino Uno that contains the sensor. The platform for the interface will not be tedious or complicated and only contain the three products seen above. To check this connection the Raspberry Pi and the Arduino will be connected by the USB. Then using the remote into Pi interface, the user must type " Is/dev/tty*" into the terminal and if "/dev/ttyACMO" is replied back, then both these interfaces are truly connected and responding to each other. Below, it is seen that dev/ttyACMO does appear amongst all of the outputs. Next, the goal is to make sure that the Arduino can output the numbers that the Raspberry Pi is outputting. If this can be done, the interface will have a successful connection test.

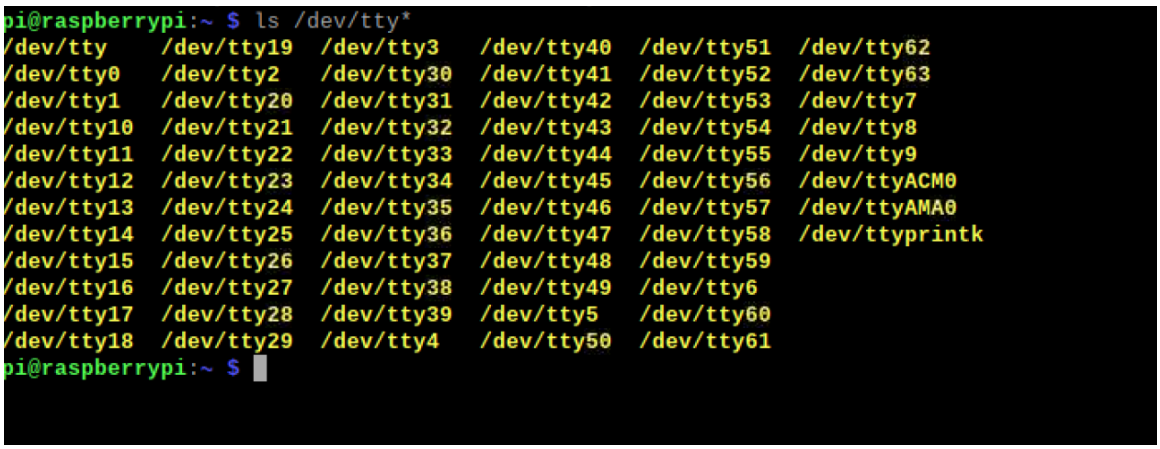


**Figure 59.** *Pi detects Arduino connection*

The easiest way for these two applications to communicate was to install Arduino IDE onto the Raspberry Pi. However, there were a lot of complications to this installation. The first installation did not go as planned. There was a mistake done in the sudo move command and an empty file was sent to the Programming panel. Then every time there was another Arduino IDE installed it would not be relocated into the Programming panel because there was an empty Arduino file already there. There were many attempts to delete this file, but it would not move and the Arduino IDE file would not open or execute. So the user decided to attempt to erase what was on the SD card and start over after hours of trying to get the Arduino Interface to properly open. So the next step was to clear the chip and restart by logging into Raspberry Pi. Before the Pi was wirelly connected to the raspberry pi and able to remote into the electronic device. After all these complications, it was best to stick to the basics and connect the Raspberry Pi to the PC and do the interface

communication here in hopes to keep the installation as simple as possible. Also the internet when wirelessly communicating into the Raspberry Pi was very slow so hopefully this may help, but if not there will be an ethernet cord added to the Pi.

After wiping the SD card and rebooting the software up, downloading Arduino IDE was super simple and it launched easily. Initially the goal for the testing was to get the Python 3 application to say hello to the arduino and for the arduino to say hello back. However, this proved to have many challenges and to be a tedious process.

So the next testing method attempted was to communicate through the red node with the arduino. Node-Red is a software that allows a very easy way to see communication between two softwares without having to do extensive coding. Installing Node-Red was not difficult and there was a sudo install that would install the Node-Red application onto the Raspberry Pi through the terminal. Using the Arduino IDE code seen below, the code basically would create a random number, and then the serial would print the data and send the Raspberry Pi interface.

The first step of the process was to ensure the Raspberry Pi and the Arduino were connected to each other via USB, open up the app Node-Red and then with the http given in the first sentence in the terminal, to open that URL which was sent to a website. After doing this, there was a serial input option which is also just our arduino( orange box) that was selected and placed on the grid. When editing the Serial Input, it was important to make sure that the correct port was utilized, in this case it needed to be /dev/ttyACM0. The other port was the mouse that was plugged into the Raspberry Pi. Once this was settled, the flow needed a debug which is the green box connected to the Arduino. After this, the user needed to debug the system and if the Arduino code was uploaded correctly to the Raspberry Pi, the Arduino would be sending numbers 0-150 to the Pi in a random order. The following system worked out perfectly and information from the Arduino was sent to the Raspberry Pi. The Arduino successfully sent information to the Raspberry Pi which shows that the Raspberry Pi will be capable of obtaining sensor information from the Arduino, and with coding will be able to turn off the webcam. Code for the one way communication from the Arduino can be found under Appendix.

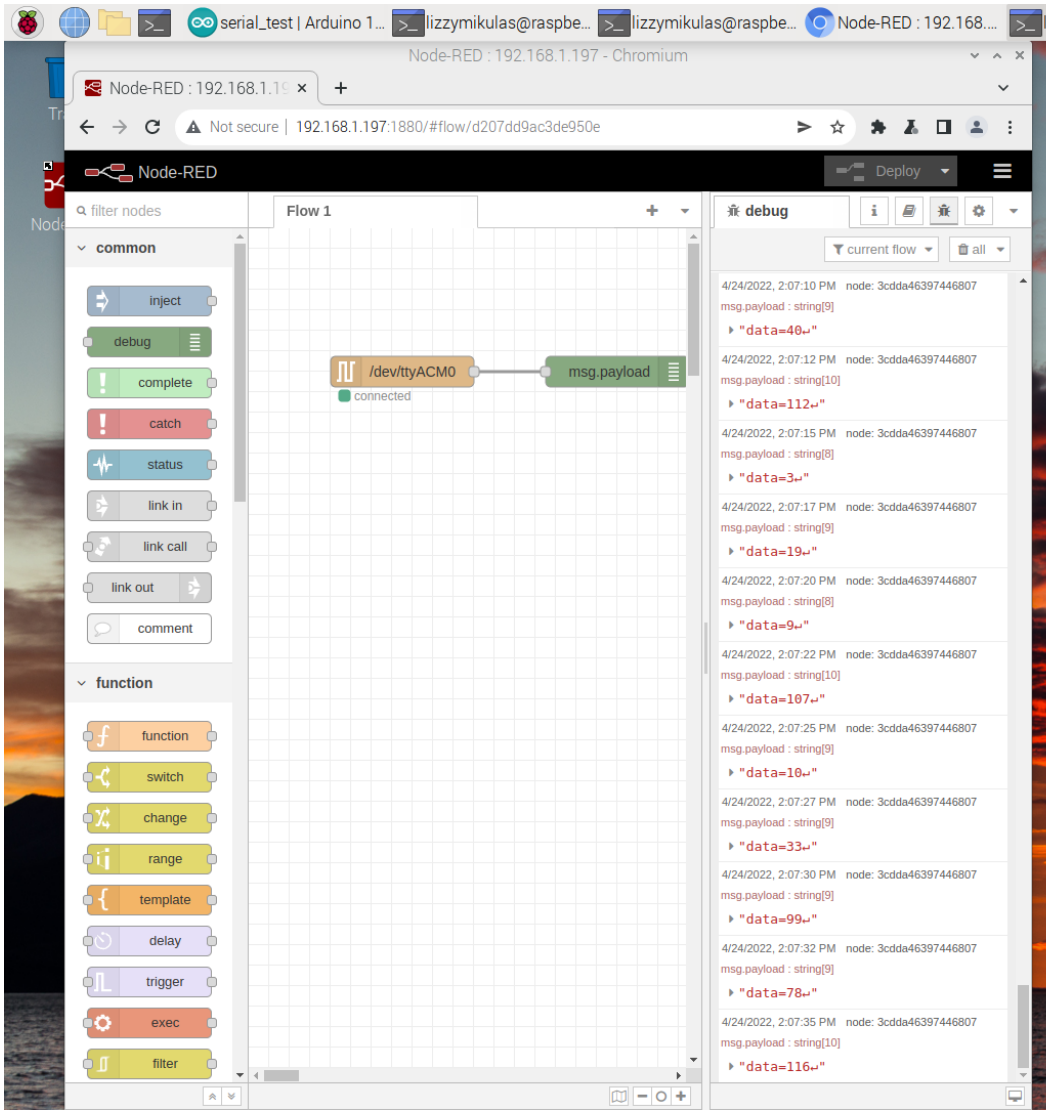


**Figure 60.** *Pi reads numbers sent from Arduino*

### *7.2.4.2 Two-way Communication Testing*

The two way communication testing consisted of python coding from the arduino that would send a string to the Arduino saying hello. The Arduino is the first string to send a message to the Raspberry Pi. Using code to send between the interfaces, the Arduino and Raspberry Pi were able to communicate to each other through the terminal of the Raspberry Pi.

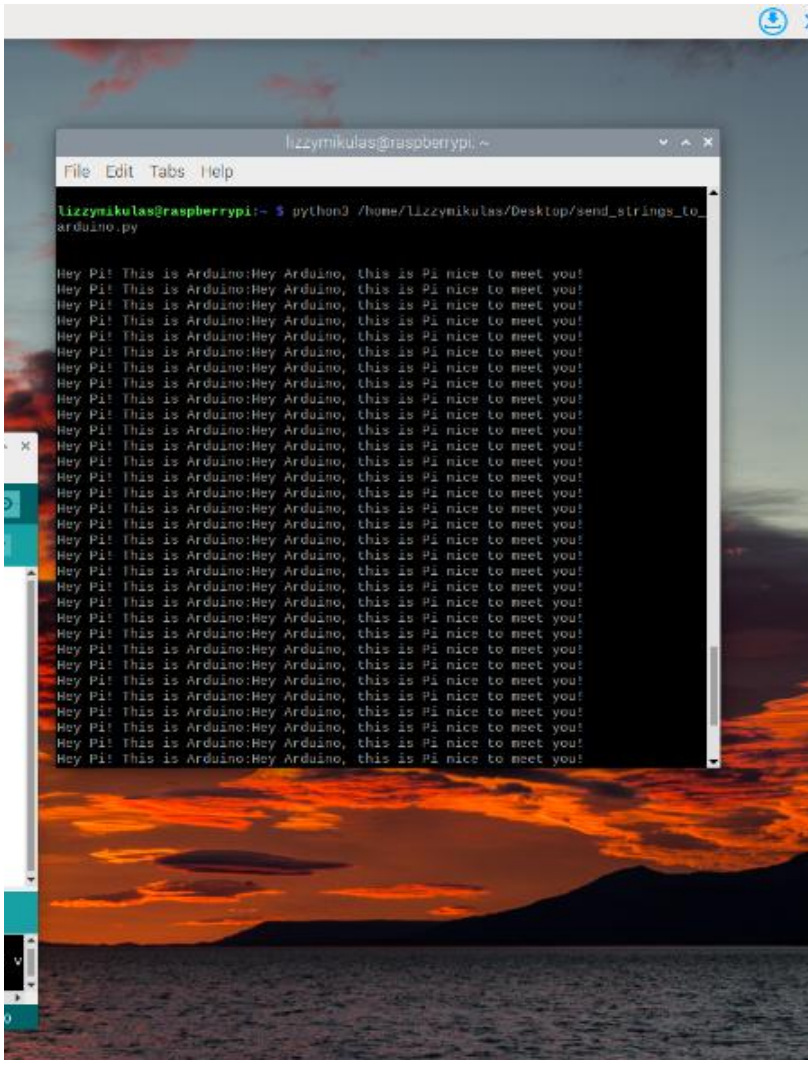


**Figure 61.** *Pi and Arduino Communication*

The Arduino is stating the first statement, and after this is received the Python which responds back with a response. The actions were delivered via chmod +x which helped execute the program. Two way communication code can be found under the Appendix.

## *7.3 Software Test Environment*

In order to fully test and verify our code, our development and testing pipeline is crucial to maintaining code quality. The testing pipeline will be composed of different components, each ensuring that the code is merged into the production code base with little to no bugs or undesirable functionality. GitHub lets us conduct conde reviews and easily grant the project with continuous integration tools with GitHub Actions. These tools and integration methodologies will aid the project's software development.

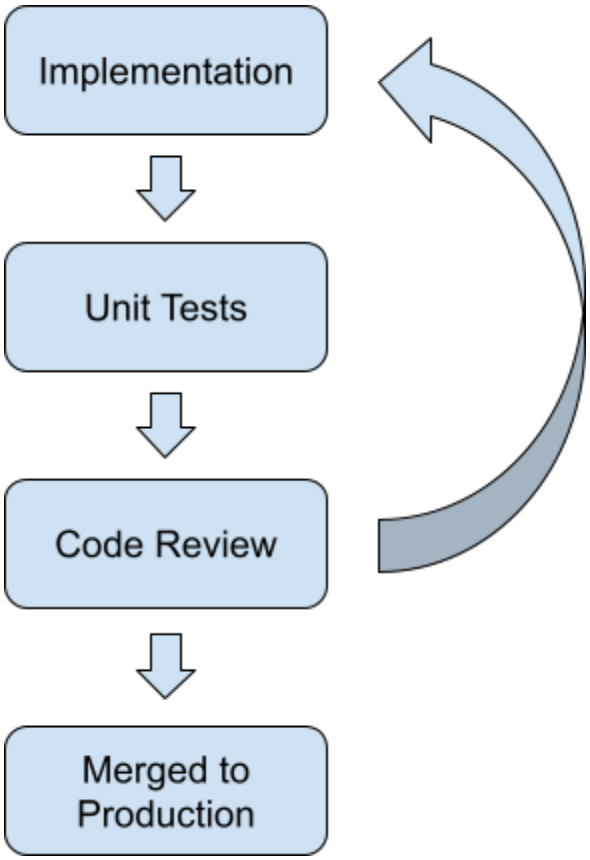


**Figure 62.** *Integration Pipeline*

## *7.3.1 Unit Tests*

Unit testing is composed of individual units of source code, sets of one or more computer program modules, along with accompanying control data, usage processes, and operating procedures tested to see whether they are fit for use within the current software [48]. Unit tests are useful for checking if the implementation work as intended in a much faster way than manual testing. These same unit tests will dictate if the new feature is ready to be merged into the production git branch. Both client and backend unit tests are discussed in detail in section 7.3.1.

Each new feature is to be placed in its branch and a pull request is to be made once the feature is completed. Merge conflicts are to be resolved by the author before the code can be merged to the production branch and all unit tests are to be passed code quality. GitHub Actions lets the project developers instruct a set of procedures to take place before the pull requests are ready to be considered for merging. These actions are set to run the testing scripts, build all the projects, and test for any runtime errors and failed unit tests.

### *7.3.2 Code Reviews*

Once all the unit tests are passed, no merge conflicts are found, and the GitHub action scripts don't throw any other error, the code can then be looked at by at least two other developers. This is when manual testing takes place and a thorough code read by these other developers would take place to ensure the functionality works as intended.

Reviewers can then write GitHub comments to recommend and/or demand changes to be made before merging completely into the codebase. The author would then follow these recommendations and edit their implementation. This process is circular as editing code may produce more bugs to be fixed, therefore it would take longer than if each individual contributor were to place their changes to the code base as needed, but this ensures minimal integration of bugs and it lets all developers know the projection and updates of the codebase.

Once at least two developers test and agree to the changes, then the code can finally be merged into the production database.

## *7.4 Software Specific Testing*

Unit tests are necessary for both the client-facing application and the backend API logic, both methodologies may be similar but the libraries used will be different due to the nature of both stand-alone applications and the different programming languages used by their frameworks.

#### *Client Unit Testing*

There are plenty of libraries that are used for unit testing javascript code, but the main problem would be the nature of front-end applications and how the developer would have to rewrite most if not all of the unit tests once a component is edited. To avoid that, the react-native-testing library allows the developer to prevent the tedious task of unit testing rewriting. react-native-testing-library is a simple library for Jest that allows you to create tests in the same way that your React Native app is used. It's based on react-testing-library, which is one of the industry standards for testing React apps. That is, your tests should be as decoupled from implementation specifics as feasible[49]. This library can be installed directly to the application through any javascript packet manager and it is widely adopted by the React Native community.

This library lets the developer test each component individually by using mock data and making sure the correct inputs and outputs are correct.

#### *API Unit Testing*

API testing is a bit more straightforward since each endpoint will be a function on its own. The authors of FastAPI suggest using their own library

called TestClient. TestClient works like most testing libraries and uses assertions that return a boolean variable dependent on the status of the test. These assertions are fairly easy to use and much better than singled out if-else statements scattered throughout the codebase. The library also contains testing functionality for HTTP body tags that will be used to communicate between both the client and API applications. Validators can also be used to validate the data models that will be used to store data to and from the API and database.

Both frontend and backend unit tests will be executed each time the project is built, committed, and merged. This will be possible with the use of GitHub Actions. These actions make sure that all unit tests are passing before a pull request can be accepted and merged to the production git branch of the application. This is extremely useful as it makes the application much safer by at least guaranteeing that the previously tested code is still working as intended after changes were made or new functionality was created. Testing through GitHub actions also take away time needed for manual testing and are extremely efficient when done correctly.

# 8. Administrative Content

In this section, administrative content such as milestone discussion and budget and finance discussion are documented. The milestone discussion outlines the pace that the team will set for both semesters and important deadlines for the report and project. The budget and finance discussion details how much the team currently has, the items necessary, and the quantity of said items. Additionally, the budget section has three tiers: the current tier, the mid tier, and the high tier. These tiers would showcase what items could be improved with a hypothetically bigger budget for the project.

## *8.1 Milestone Discussion*

The initial project milestone for this semester is to understand what components and materials we will need to start our project as well as making sure we have all the necessary components. The initial project milestone for next semester is to have a working prototype of a human-following robot via motion sensor and/or mobile application. Table 4 below will show the project milestones for the first semester, while Table 5 will show the project milestones for the second semester of senior design.

| # | Date | Description |
|---|---|---|
| 1 | February 4 | Divide and Conquer Document (10 pages) |
| 2 | February 18 | Updated Divide and Conquer (20-25 pages) |
| 3 | February 28 | Have sponsorship and/or IP access |
| 4 | March 25 | Draft Senior Design 1 Documentation (60 pages) |
| 5 | April 8 | Draft Senior Design 1 Documentation (100 pages) |
| 6 | April 26 | Submit final revision Senior Design 1 Documentation (120 pages) |
| 7 | April 26 | Have completed project design |
| 8 | May 1 | Order all necessary components |

| # | Date | Description |
|---|---|---|
| 9 | May 16 | Start software development |
| 10 | May 31 | Have all necessary components |

**Table 24.** *Semester 1 Project Milestones*

| # | Date | Description |
|---|---|---|
| 11 | August 15 | Start second semester |
| 12 | August 15 | Start hardware assembly |
| 13 | August 15 | Have software application prototype complete |
| 14 | August 15 | Start housing assembly for 3D Hologram |
| 15 | September 15 | Complete Assembly of 8, 11, 12 |
| 16 | September 24 | Complete Unit Testing of 8, 11, 12 |
| 17 | September 30 | Complete Integration First Pass |
| 18 | October 8 | Complete Integration Second Pass (if necessary) |
| 19 | October 11 | Complete Integration Third Pass (if necessary) |
| 20 | November 16 | Complete Integration Testing (end to end testing) |
| 21 | November 18 | Project Completion |
| 22 | Nov 30 | Committee Presentation |
| 23 | Dec 2 | Senior Design Expo |

**Table 25.** *Semester 2 Project Milestones*

## *8.2 Other Projects*

The below table outlines the top three projects the team considered doing at the start of the Senior Design 1 semester. The 4 criteria we used to judge and compare each of the projects was the estimated cost of the project, our motivation to complete said projects, the technological goals that fit the requirements for the class, as well as the difficulty of each project given our time constraint.

| | Cost | Motivation | Technological Goals | Difficulty/Time Constraint | Score |
|---|---|---|---|---|---|
| Weight of Each Criteria | 3 | 5 | 4 | 2 | |
| Project 1 (Holomon) | **1*3 = 3** | **3*5 = 15** | **3*4 = 12** | **1*2 = 2** | **32** |
| Project 2 (Ring Facial Security) | 2*3 = 6 | 2*5 = 10 | 2*4 = 8 | 3*2 = 6 | 30 |
| Project 3 (Smart Fridge) | 2*3 = 6 | 2*5 = 10 | 1*4 = 4 | 2*2 = 4 | 24 |

**Table 26.** *Decision Matrix of Other Projects*

## *8.3 Budget and Finance Discussion*

Our project will have a smaller or comparable budget to most senior design projects which heavily influences the type of display technology we can use to produce our prototype. The project will be both self-funded of $100 from each team member and are looking into receiving sponsorship from Niantic and/or sole proprietary sponsors. The minimum budget this project will have is therefore $400. We may also have a single-affiliated sponsor who is interested in donating another $100 for a total of $500 if Niantic declines to sponsor.

Table 27 below shows the updated price per unit of each component necessary in our project along with the quantity and the actual expense for each item. The last row in the table outlines the total expense the team used over the set budget. This is the complete bill of materials for the final product. Tables 28 and 29 are mid-tier and high-tier budgets that would allow the hologram to look nicer given a higher financial budget, since the team would not need to rely on Pepper's Ghost.

| Components | Price Per Unit | Quantity | Expense |
|---|---|---|---|
| Raspberry Pi Model B + power cable | $200.00 | 1 | $0.00 |
| TV + power cable | $100.00 | 1 | $0.00 |
| ATMEGA328p chip | $15.50 | 2 | $31.00 |
| PCB manufacturing | $0.95 | 5 | $4.75 |
| Medium Density Fiberboard | $28.00 | 4 | $112.00 |
| Tempered Hardboard | $8.50 | 1 | $8.50 |
| Black spray paint | $6.50 | 3 | $19.50 |
| Matte varnish spray paint | $6.50 | 2 | $13.00 |
| Plexiglass | $24.00 | 2 | $48.00 |
| Logitech HD Pro Webcam C920 | $70.00 | 1 | $0.00 |
| CP2102 USB to TTL module | $7.50 | 1 | $7.50 |
| HC-SR501 PIR Sensor | $8.00 | 1 | $8.00 |
| 9V Battery | $4.00 | 1 | $4.00 |
| HDMI to micro-HDMI cable | $9.00 | 1 | $9.00 |
| Acrylic glue | $9.50 | 1 | $9.50 |
| 1 ¾” wood screws | $10.00 | 1 | $10.00 |
| Crystal Oscillator | $10.00 | 1 | $10.00 |
| Power Jack Solder Component | $1.40 | 5 | $7.00 |
| 22pF ceramic capacitor | $0.23 | 1 | $6.00 |
| 2.54mm 40P jumper ports solder | $0.80 | 1 | $8.00 |
| Voltage regulators | $10.00 | 1 | $10.00 |
| 9V battery clip | $0.70 | 1 | $7.00 |
| Work gloves + goggles | $30.00 | 1 | $30.00 |
| Meter stick, tape measure, sharpie | $29.00 | 1 | $29.00 |
| **TOTAL EXPENSE / BUDGET** | | **$363** | **$400** |

**Table 27.** *Project Budget and Expenses (Current Tier 1)*

| Item | Expected Quantity | Budget Per Allocated Item |
|---|---|---|
| CP2102 USB to TTL Cable | 1 | $7 (acquired) |
| HC SR501 Sensor | 1 | $8 (acquired) |
| Power Supply (9V battery) | 2 | $10 |
| **Item** | **Expected Quantity** | **Budget Per Allocated Item** |
| HDMI to micro-HDMI Cable | 1 | $11 (acquired) |
| PCB | 1 | $15 |
| Acrylic Cement / Glue | 1 | $20 |
| Plexiglass | 1 | $100 |
| Wood | 4 | $129 |
| Autostereoscopic 3D Display | 1 | $1200 |
| Raspberry Pi 4 Module B | 1 | $0 (acquired) |
| Arduino Uno | 1 | $0 (acquired) |
| Webcam | 1 | $0 (acquired) |
| TV | 1 | $0 (acquired) |
| **Current Estimated Total** | | **$1500** |
| **Budget Total** | | **$1500** |

**Table 28.** *Project Budget and Expenses (Mid Tier 2)*

| Item | Expected Quantity | Budget Per Allocated Item |
|---|---|---|
| CP2102 USB to TTL Cable | 1 | $7 (acquired) |
| HC SR501 Sensor | 1 | $8 (acquired) |
| Power Supply (9V) | 5 | $10 |
| HDMI to micro-HDMI Cable | 1 | $11 (acquired) |
| PCB | 1 | $15 |
| Acrylic Cement / Glue | 1 | $20 |
| Plexiglass | 1 | $200 |
| Wood | 1 | $229 |
| Autostereoscopic 3D display | 1 | $4500 |
| Raspberry Pi 4 Module B | 1 | $0 (acquired) |
| Arduino Uno | 1 | $0 (acquired) |
| Webcam | 1 | $0 (acquired) |
| TV | 1 | $0 (acquired) |
| **Current Estimated Total** | | **$5000** |
| **Budget Total** | | **$5000** |

**Table 29.** *Project Budget and Expenses (High Tier 3)*

In summary, the team will be using the items and expenditure from tier 1 with cheaper material of different items used to create HoloMon.

# 9. Conclusion

In this chapter, we explore the team's final regards to HoloMon by addressing future iterations of the project, the marketability, and addressing the team's concluding thoughts on HoloMon.

## *9.1 Future Iterations*

We have a few suggestions from our own design that could be improved in future iterations. Most of these were not included in the first iteration due to availability, financial, and time resources.

The first of them is introducing a "rover" based foundation for the hologram that actually follows a user around and has collision detection (as briefly mentioned in the stretch goals of chapter 2). The next would be to make the HoloMon more portable for marketability by making the overall project smaller. The third main improvement that could be made in future iterations is using something more like an autostereoscopic 3D display so that the viewing angle can be seen 360 degrees around as well as having the Pokémon be seen in broad daylight.

With these iterations, it is possible to make HoloMon follow the initial motivating idea of it being an entertainment system for children to learn how to socialize from a socially acceptable distance.

## *9.2 Marketability*

As aforementioned, the project was never designed with marketability in mind but rather how our budget, time, and research would influence our decisions. The biggest setback for HoloMon becoming marketable is its lack of mobility. Essentially, for anything like HoloMon to be desirable on a marketable level, its usefulness or mobility would have to be on par with that of smartphones.

While this iteration was overall not as expensive as the mid to high tier budgets, it is still a pretty costly device. There could be a significant drop in price if some components, such as the TV were downsized to a tablet or monitor while removing the intermediary microcontroller unit could eliminate some costs as well.

## *9.3 Concluding Thoughts*

We successfully researched, designed, and tested a most likely novel entertainment system called HoloMon. By having each component and testing their subsystems, we have fulfilled our academic requirements while inspiring further interest in the fields of computer engineering and electrical engineering.

# Appendix A - Copyright Permissions

**The Pokémon Company (International)**

The Pokémon Company (International) has allowed us to use IP access as long as we are not profiting off their IP. Since our project is solely for the purposes of education, we have deemed HoloMon safe to use The Pokémon Company (International)'s assets.

# Appendix B - References


[1] Jason Geng, Three-dimensional display technologies." Adv Opt Photonics,vol. 5, no. 4, pp. 456–535, Dec. 2012, doi: undefined. https://www.ncbi.nlm.nih.gov /pmc/articles/PMC4269274/ (accessed Feb. 24, 2022).

[2] M. Oudah, A. Al-Naji and J. Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," J Imaging., vol. 6, no. 8, pp. 73, Aug. 2020, doi: undefined.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC832 1080/ (accessed Mar. 23, 2022).

[3] "Stack Overflow Developer Survey 2021,". https://insights.stackoverflow.com/survey/2021#most-popular-technologies-platfo rm-prof (accessed Mar. 24, 2022).

[4] "What are different holographic display technologies in the market? - HOLOFIL," . https://www.holofil.com/post/2018/11/21/is-holofil-a-3d-hologram-3d-Holographic-display-what-other-3d-display-techniques-are-there (accessed Mar. 23, 2022).

[5] "PEP 8: The Style Guide for Python Code,". https://pep8.org/ (accessed Mar. 24, 2022).

[6] M. Smallcombe, "SQL vs NoSQL: 5 Critical Differences," *Integrate.io*, Jul. 23, 2021.https://www.integrate.io/blog/the-sql-vs-nosql-difference/#:~:text=SQL%20d atabases%20are%20vertically%20scalable,data%20like%20documents%20or% 20JSON. (accessed Mar. 24, 2022).

[7] "React Native Coding Standards and Best Practices | by Gilshaan Jabbar | Medium,".https://gilshaan.medium.com/react-native-coding-standards-and-best-p ractices-5b4b5c9f4076 (accessed Mar. 23, 2022).

[8] "MongoDB Naming Standards and Guidelines | TheCodeBuzz,". https://www.thecodebuzz.com/mongo-db-naming-conventions-standards-guidelines/ (accessed Mar. 23, 2022).

[9] A. Mahmud, “NoSQL Databases: When to Use NoSQL vs SQL,” *ServerWatch*, Jan. 19, 2022. https://www.serverwatch.com/guides/when-to-use-nosql/ (accessed Mar. 24, 2022).

[10] D. Bruce, “Understanding the Pros and Cons of MongoDB,” *knowledgenile*, Apr. 21, 2021. https://www.knowledgenile.com/blogs/pros-and-cons-of-mongodb/ (accessed Mar. 24, 2022).

[11] Raspberry Pi Arduino Serial Communication - Everything You Need To Know - The Robotics Back-End, 2022. https://roboticsbackend.com/raspberry-pi-arduino-serial-communication/#:~:text=The%20easiest%20way%20is%20to%20use%20a%20USB,computer%20%28with%20the%20Arduino%20IDE%29%20to%20your%20board. (accessed Mar 23, 2022).

[12] “Native vs Cross-Platform Development: Pros & Cons Revealed,” *Uptech.team*, 2016. https://www.uptech.team/blog/native-vs-cross-platform-app-development (accessed Mar. 24, 2022).

[13] Wikipedia Contributors, “Parallax barrier,” *Wikipedia*, Feb. 01, 2021. https://en.wikipedia.org/wiki/Parallax_barrier (accessed Mar. 24, 2022).

[14] “UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter,” *Analog.com*, 2020. https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html#:~:text=By%20definition%2C%20UART%20is%20a,going%20to%20the%20receiving%20end. (accessed Mar. 24, 2022).

[15] “I2C Primer: What is I2C? (Part 1),” *Analog.com*, 2022. https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-1.html (accessed Mar. 24, 2022).

[16] “Introduction to SPI Interface,” *Analog.com*, 2018. https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html (accessed Mar. 24, 2022).

[17] Wikipedia Contributors, “Holography,” *Wikipedia*, Feb. 07, 2022. https://en.wikipedia.org/wiki/Holography (accessed Mar. 24, 2022).

[18] Wikipedia Contributors, “Pepper’s ghost,” *Wikipedia*, Mar. 12, 2022. https://en.wikipedia.org/wiki/Pepper%27s_ghost (accessed Mar. 24, 2022).

[19] “Pepper’s Ghost: Hologram Illusion - Science World,” *Science World*, Jul. 27, 2020. https://www.scienceworld.ca/resource/peppers-ghost-hologram-illusion/ (accessed Mar. 24, 2022).

[20] "MSP430G2553", Ti.com, 2022. [Online]. Available: https://www.ti.com/product/MSP430G2553. [Accessed: 24- Mar- 2022].

[21] "MSP430FR6989", Ti.com, 2022. [Online]. Available: https://www.ti.com/product/MSP430FR6989. [Accessed: 24- Mar- 2022].

[22] *Raspberrypi.org*, 2017. https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/1 (accessed Mar. 25, 2022).

[23] Wikipedia Contributors, “Transparency meter,” *Wikipedia*, Feb. 19, 2022. https://en.wikipedia.org/wiki/Transparency_meter (accessed Mar. 25, 2022).

[24] "Arduino Uno Rev 3", Docs.arduino.cc, 2022. [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3. [Accessed: 29- Mar- 2022].

[25] "MSP-EXP430G2ET", 2022. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/MSP-EXP430G2ET/9608004?s=N4IgTCBcDaILIGUAKAWAzABgOJgKy7RAF0BfIA. [Accessed: 29- Mar- 2022].

[26] "Logitech Fluid Crystal Technology HD Webcam, 720p," https://www.dontwasteyourmoney.com/products/logitech-fluid-crystal-technology-hd-webcam-720p/.

[27] Wikipedia Contributors, “Kinect,” *Wikipedia*, Mar. 27, 2022. https://en.wikipedia.org/wiki/Kinect (accessed Apr. 04, 2022).

[28] “C Coding Standard,” *Cmu.edu*, 2019. https://users.ece.cmu.edu/~eno/coding/CCodingStandard.html

[29] “Электронный портал | Datasheets, Микроконтроллеры msp430 avr pic mega128 microcontroller Flash-memory,” *kazus.ru*. http://kazus.ru/nuke/modules/Downloads/pub/147/0/IPC-2221%20Amd1.pdf (accessed Apr. 05, 2022).

[30] "Holographic displays, pyramid, 3d hologram display - Olomagic," https://www.olomagic.com/.

[31] “HC-SR501 PIR MOTION DETECTOR.” [Online]. Available: https://www.mpja.com/download/31227sc.pdf

[32] “HC-SR501 Passive Infrared (PIR) Motion Sensor.” Accessed: Apr. 07, 2022. [Online]. Available: https://www.epitran.it/ebayDrive/datasheet/44.pdf

[33] “ROBOT . HEAD to TOE Product User’s Manual -HCSR04 Ultrasonic Sensor User’s Manual ROBOT . HEAD to TOE Product User’s Manual -HCSR04 Ultrasonic Sensor Index,” 2013. [Online]. Available: https://web.eece.maine.edu/~zhu/book/lab/HC-SR04%20User%20Manual.pdf

[34] Light Field Lab, “Light Field Lab,” *Light Field Lab*, 2018, https://www.lightfieldlab.com/.

[35] “Pokémon holograms with image tracking,” *www.youtube.com*. https://www.youtube.com/watch?v=p57ZHnR0QlE (accessed Apr. 07, 2022).

[36] Blogger, “Official screenshots for petting, tickling and feeding Eevee and Pikachu in Pokémon Let’s Go Pikachu and Let’s Go Eevee,” *Pokémon Blog*, Jul. 12, 2018. https://Pokémonblog.com/2018/07/12/official-screenshots-for-petting-tickling-and-feeding-eevee-and-pikachu-in-Pokémon-lets-go-pikachu-and-lets-go-eevee/ (accessed Apr. 07, 2022).

[37] “Amazon EC2,” *Amazon Web Services, Inc.*, 2020. https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bced9843&sc_channel=ps&sc_campaign=acquisition&sc_medium=ACQ-P|PS-GO|Brand|Desktop|SU|Compute|EC2|US|EN|Text&s_kwcid=AL!4422!3!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQjwl7qSBhD-ARIsACvV1X086SDlLsIcBKCLN-98Sjv-Fp6bzDmUMPujrO23MV0KvY334coeKT4aAgwyEALw_wcB:G:s&s_kwcid=AL!4422!3!467723097970!e!!g!!aws%20ec2 (accessed Apr. 07, 2022).

[38] https://www.facebook.com/hashedoutssl, “The difference between Encryption, Hashing and Salting,” *Hashed Out by The SSL Store™*, Dec. 19, 2018. https://www.thesslstore.com/blog/difference-encryption-hashing-salting/ (accessed Apr. 07, 2022).

[39] Wikipedia Contributors, “Model–view–controller,” *Wikipedia*, Mar. 22, 2022. https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller (accessed Apr. 07, 2022).

[40] Wikipedia Contributors, “JSON Web Token,” *Wikipedia*, Dec. 19, 2021. https://en.wikipedia.org/wiki/JSON_Web_Token (accessed Apr. 07, 2022).

[41] Wikipedia Contributors, “Connection string,” *Wikipedia*, Jan. 03, 2022. https://en.wikipedia.org/wiki/Connection_string (accessed Apr. 07, 2022).

[42] “AsyncStorage · React Native,” *Reactnative.dev*, Mar. 30, 2022. https://reactnative.dev/docs/asyncstorage (accessed Apr. 08, 2022).

[43] Wikipedia Contributors, “Electronic waste,” *Wikipedia*, Mar. 29, 2022. https://en.wikipedia.org/wiki/Electronic_waste (accessed Apr. 08, 2022).

[45] C. Atwell, "The top microcontroller boards - Embedded.com", Embedded.com, 2022. [Online]. Available: https://www.embedded.com/the-top-microcontroller-boards/. [Accessed: 08- Apr- 2022].

[46] "Chip shortage: how the semiconductor industry is dealing with this worldwide problem", World Economic Forum, 2022. [Online]. Available: https://www.weforum.org/agenda/2022/02/semiconductor-chip-shortage-supply-chain/. [Accessed: 08- Apr- 2022].

[47]"The Raspberry Pi 1, 2 and 3 compared", Ste Wright, 2022. [Online]. Available: https://www.stewright.me/2016/03/raspberry-pi-1-2-3-compared/. [Accessed: 25- Apr- 2022].

[48] Tri Basuki Kurniawan, “Unit Testing and Continuous Integration (CI) using Github Actions,” *Medium*, 2021. https://medium.com/thelorry-product-tech-data/unit-testing-and-continues-integration-ci-in-github-action-for-python-programming-c8ad57fae3a1 (accessed Apr. 25, 2022).

[49] Matthieu Alingrin, “How to Test Your React Native App with React-native-testing-library,” *Bam.tech*, Sep. 18, 2019. https://blog.bam.tech/developer-news/how-to-test-your-react-native-app (accessed Apr. 25, 2022).

[50] “PuTTY,” *Wikipedia*, Feb. 25, 2020. https://en.wikipedia.org/wiki/PuTTY

# Appendix C - Code

## *C.1 One Way Communication Code*

```
void setup() {
Serial.begin(9600);                //Serial
}


void loop() {
  int data = random(0,150);
  Serial.print("data:");
  Serial.println(data);
```

```
  delay(2010);                    // Delay
}
```

Two Way Communication Code: