Backup Camera System for Power Chairs

by

Emily Becker, Elissa Desmann, Vishnu Pillai, Nicholas Robinson, Aidan Rogers
Biomedical Engineering, George Washington University Student Expected Graduation May 2019

A Senior Capstone Project Overview submitted to

The Faculty of Biomedical Engineering The School of Engineering and Applied Science of the George Washington University in partial fulfillment of the requirements for the degree of Bachelor of Science

May 6, 2019

Abstract

The goal of this capstone project was to design, create, and test a backup camera system for a power chair. The team researched several backup camera systems already on the market and chose components that would act together as a complete, useable device for our client and other clients using a variety of mobility devices. Several subsystems were designed including a depth camera, a display, a microcontroller, a power supply circuit, and encasements housing these components.

The user has the ability to view the environment behind him with or without a visual pathway for guidance regarding the distance of any objects behind him. A visual pathway with distances of 0.75m, 1.5m, 3.5m, and 5.0m will give him an accurate benchmark to determine an object's distance from the back of his chair. On the back of the chair, within an encasement, the camera takes frames from the environment and transfers them to the microprocessor for further processing and/or display depending on user preference. The data is then transferred to the monitor for display. This monitor is secured to the user's chair via a rod that is fastened to the side of the user's seat, which allows it to be adjusted for a proper viewing angle. The entire system is powered by a power supply circuit that functions to convert battery power into a voltage-regulated DC power output.

These subsystems were tested for frame rate, depth accuracy, latency, electrical safety, and material strength, respectively. The team succeeded in designing an accurate display of environment with depth thresholds with more than 95% accuracy, exceeding our design specifications. The team also succeeded in minimizing system latency, although it never quite fell below our specified value of 300ms. The encasements used to house our components were tested for tensile strength and elongation at break in order to determine if the chosen ABS material was strong and durable enough to support their weight.

The total cost of the project was less than $250 as intended. Future work can be done to enhance this system. Further functions might include a rotating visual pathway, audible alerts, a portable and rechargeable power supply, and reduced camera latency can improve the overall device performance.

Contents

LIST OF FIGURES

LIST OF TABLES

1. Background

1.1. Patient Population

There are an estimated 3.3 million wheelchair and power chair users in the United States [1]. Of those users, only a small fraction has access to backup assistance systems, making reverse mobility dangerous and often tedious. Most backup camera adaptations for mobility devices are expensive and hard to come by on the market, costing customers upwards of $250 [2], and are typically not compatible with all types of mobility devices. To remedy this issue for wheelchair, power chair, scooter, and other mobility device users across the world, our team will design a backup assistance system that will be able to follow the user from chair to chair, no matter where in the world they want to go. Our goal is to give our clients the freedom they deserve and to ensure our product keeps them safe wherever they may go. To make our design accessible to people of all backgrounds and diagnoses, our team will need to simplify the subsystems in a way that ensures safe, easy, affordable, and effective use for all potential clients.

1.2. Overview of Project Goals

Our main client is Dr. Robert Carroll, a retired electrical engineering professor at the George Washington University who has damaged nerves in his left ankle that resulted from a surgery that went awry. For the past five years, Dr. Carroll has been able to regain his independence with the use of a rear-wheel, model C350 power chair purchased from Permobil [3]. While this chair helps immensely in aiding Dr. Carroll's mobility, it does not come with any rear-view camera detection system, leaving our client unable to see the environment behind him. All of the clients we have interviewed have expressed the need for the following assistive features that would aid them in reverse mobility:

- The user will be able to view the environment behind them.
- The user will be able to gauge the depths of objects behind them.
- The user will be alerted when objects are too close to the back of their chair.
- The user will be able to adjust the display for a preferred viewing angle.
- The user will be able to use the backup device indoors and outdoors.

1.3. Objectives, Functions, and Specifications

User Story	Objectives & Functions	Specifications
View the environment behind them	Image acquisition (observe the environment behind the chair)	Pixel density of at least 200ppi +/- 100ppi for viewing distance of 40-50cm [4]
	Data transfer and display (display the environment to the user)	Avg. camera-to-display latency < 300ms
Gauge the depths of objects behind them	Device must have a visual pathway to guide the user	Visual pathway must accurately reflect distance thresholds
	Measure the distance of obstacle to chair with high accuracy	Distance must be measured over 95% accuracy (actual vs calculated distance)
Be alerted when objects are too close to the back of their chair	Device must have an alarm system for approaching obstacles	Volume should be preference of the user [5]
	Alert the user of obstacles via audio cue (beep)	Sound should be between 150-1000Hz, with four harmonics 300 to 4000 Hz each [5]
Adjust the display for a preferred viewing angle	Monitor must be adjustable and supported	Mount with arm that can be rotated and support the weight of monitor with 150% safety margin
Use the backup device indoors and outdoors	Power source must be portable, rechargeable, and protected from the outside environment	- Battery capacity < 100Wh [6] - Non-spillable, dry or gel battery [6] - Battery life >= 4 hours of continuous use - Charging time < 12 hours (overnight) [7]

Table 1.3.1: Objectives, functions, and specifications of the backup camera for power chair

1.4. Existing Products

There are some backup camera adaptations that already exist on the market, but these products are either not universally compatible for all types of mobility devices, expensive, or limited in scope and function. Our backup camera device will differ by being compatible with wheelchairs, power chairs, scooters, and all other types of mobility devices. Additionally, our device will be no more costly than existing products (under $250) and simple to assemble and navigate so users of all ages, afflictions, and socioeconomic classes can benefit from it.

1.5. Solution

Our solution to the lack of user visibility when reversing is to design a portable, rechargeable backup camera system complete with a touch-screen monitor that will attach to our client's power chair and display the environment behind the client's chair. Additionally, a depth camera will be attached to the back of our client's chair along with a microcontroller that will process the data captured by the camera. These components will be powered by a power supply circuit that will attach to a rechargeable battery. All subsystems of this device will be encased in 3D printed ABS material that will secure the equipment to our client's chair and preserve it from environmental factors.

Some challenges that may occur are: finding an appropriate battery type that complies with air travel regulations while providing the high amount of power needed to power the system, implementing encasement material that can withstand the temperature of the power circuit and its components while protecting electrical circuits from the outside environment, and creating an algorithm that extracts and displays the data captured from the depth camera with minimal latency.

1.6. Definitions of Success

The definition of success is a portable backup camera for a power chair that takes input from a camera on the back of the chair and displays it on a monitor attached to the arm of the chair. Features of this device include encasements for individual components such as a rotating monitor mount and a power supply that can power the entire system with 4 hours of continuous use. Additional attributes include a visual pathway that accurately reflects the distance of objects behind the chair and a beep that sounds when objects are close to the back of the chair. A camera acquires data like images and transfers it to the microprocessor. The user must be able to generalize an accurate distance of an object behind his chair through the use of visual aids and an audible alert system. These are implemented through coding and functions stored in a microprocessor. The monitor, stored in an encasement on a mount, is adjustable for the user's optimal usage and the power source is rechargeable, portable, and protected from the outside in another encasement.

2. Technical Justification

2.1. Technical Overview

The subsystems of our device consist of a camera-to-UP Board system, a monitor that display the visual pathway, a portable and rechargeable power supply, and encasements for the monitor and other components. The connection of systems is straightforward, as described below, and would likely be put together by the user upon receiving the device.

The first subsystem initializes the device's capabilities through the RealSense Depth Camera D435 acquiring images of the environment behind the user as data and transferring it to an Aaeon UP Board for display. These are connected with a USB-C (to the camera) to USB-3 (to the UP Board) cable. The UP Board is powered through a 5V/5A power supply. An executable C++ file on the UP Board (either rs-capture or rs-imshow, depending on if the user wants a visual pathway), when run, allows this data to be transferred over and displayed or further modified for display.

The second subsystem displays the visual pathway on top of the pipelined images from the RealSense Depth Camera D435. The code written permanently modifies pixel color in the shape of the visual pathway on the display/monitor. This visual pathway gives a reference, to the user, for objects distance on the screen.

The third subsystem supplies the specified amount of voltage and current to the overall system, which is a maximum of 5V and 4A of current. The battery implemented must be capable of supplying these 20W of power consistently for a period of at least 4 hours, and it must comply with air travel regulations, which state that batteries in cargo must be made of non-spillable gel or dry material and produce less than 100Wh of energy.

The encasement is capable of encasing an Intel Realsense D435 camera and UP Board. The monitor case can holster the monitor and can be mounted. The intended use of these embodiments allows the user to observe the environment behind them.

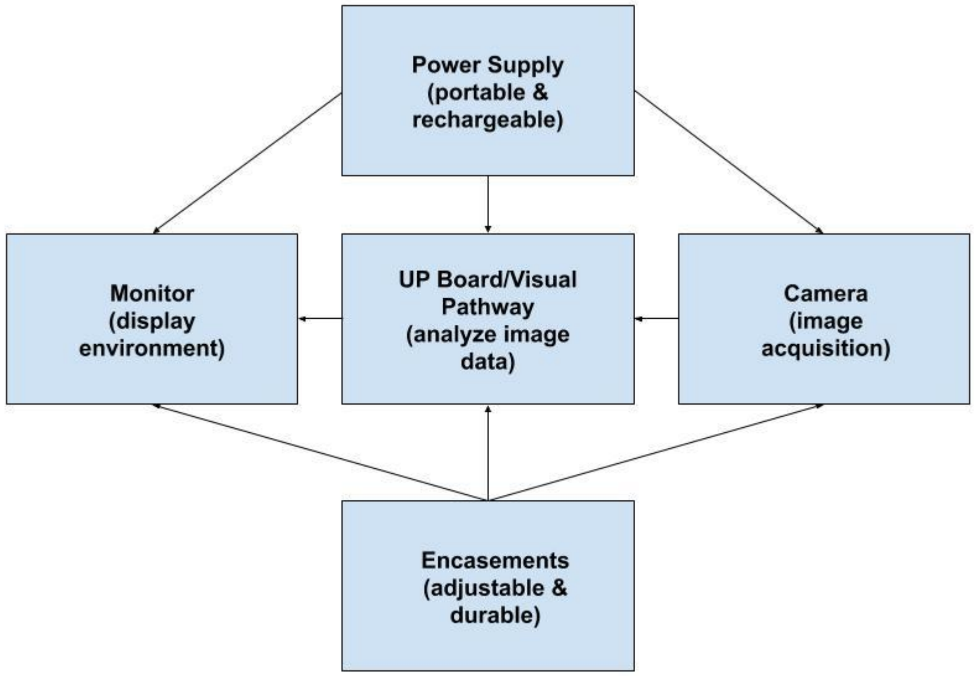


Figure 2.1.1: *Functional block diagram of subsystems, objectives, and functions*

2.1.1. Subsystem 1: Camera and UP Board

This subsystem initiates data transfer from the RealSense camera to the UP Board and processes the incoming frames in preparation for display. Two main codes will be used by the client on the UP Board, one called rs-imshow.cpp [11] and another called rs-capture.cpp [10]. Both can be found in the Github repository "librealsense" [14] made specifically for the RealSense camera. These codes are nearly identical in output, but have different functions implemented within their codes. Both files, when run, show a livestream obtained from the display of the environment behind the user that he can use while he is reversing. Rs-imshow displays the same environment but with a visual pathway overlay as explained in section 2.1.2.

The primary function of this subsystem is to acquire the data that holds the image frames and prepare them for display. Through initiating a pipeline from librealsense in C++ on a Windows-imaged processor, framesets can move from the camera to the UP Board to be modified or transferred straight to the output, a monitor for the user to view the environment.

Both codes use the same functions that perform nearly identically in testing to transfer data to be processed and displayed. They differ in display due to the complexity of rs-imshow. This function implements opencv, another Github repository that uses a computer vision library. This allows a coded-in modification of images as the file runs, which replaces individual pixels on the screen with colors to create the appearance of a visual pathway, similar to that of a car backup system. More on the visual pathway's use, display, and accuracy is explained in section 2.1.2.

Some of the technical challenges with this subsystem began with the correct operating system and language to code in. Initially, the team anticipated a finished product in Python on a Raspberry Pi (with an OS in Raspbian), but after some research on the advantages and disadvantages of different OS and languages, the most user-centric and easiest system would be one with C++ and Windows. The process to download and build the repositories in the correct format was time-consuming but has worked the best. Other technical difficulties included dissecting and studying the repositories' codes to modify them with some ease. Through technical reports, however, this problem was solved by creating diagrams and defining the different functions called within the files.

Commented codes for both rs-imshow.cpp and rs-capture.cpp as well as flow charts explaining the codes, can be seen in Appendix 5.3.1-5.3.4.

2.1.2. Subsystem 2. Display/Visual Pathway

This subsystem utilizes an edited version of rs-imshow.cpp [11] and displays a visual pathway to the user on every frame transferred to the monitor display. This visual pathway can be viewed by the user on the 7-inch monitor that will sit near the left armrest of the power chair. The display had to meet the specification of pixels per inch 200 pixels +/- 100 pixels at a distance between 40 and 50 centimeters. The visual pathway is divided into three separate sections, red, yellow and green. Each section is trapezoidal and defines a region of space perpendicular to the user's power wheelchair. Furthermore, each section is broken up by lines that show known distance values in the corner.

The purpose of the monitor in the project is substantial as it will show the background environment to the user. The ability to continually see the rearview allows the user to maneuver around obstacles, thus ensuring safety. The challenges that came with this subsystem was guaranteeing visibility from a certain distance since the screen will not be directly in front of the user. Also, it is essential to ensure that the latency is minimal to protect the user's safety. To test the visibility of the display, a survey was conducted, where the participants were asked if they could view the display correctly and accurately from the distances stated in the specifications. A z-test was conducted following the survey to statistical analyze the survey data. For latency, the camera was pointed at a timer on a laptop in real time for over a minute. A video was taken of the entire process, including the laptop's screen and the monitors' screen. The video was paused at different intervals to compare the laptop's timer reading and the monitor's timer reading. The difference between the two values gave the latency period.

To draw the pathway, a series of for loops are used in cohesion with the function vec3b [20]. This is an opencv[15] function that takes a vector of three bytes, each an unsigned character. The character values are between 0 and 255 and correspond to color intensities for Red, Green, and Blue. The for loops were used to alter the color of each pixel selected on the screen to a specific color for a given frameset; in this case every frameset. It is shown in Figure 5.3.2.

When backing up the visual pathway will provide the user with a reference for how far away objects are on the screen. The red subsection defines the area closest to the user, spanning from directly behind the power chair to 1.5 meters away, at 0.75 meters an additional line was drawn to assist the user in identifying objects extremely close to them. The largest region is closest to the user to bring their attention to objects directly behind them. The yellow-defined region spans from 1.5 meters to 3.5 meters. The green-defined region is the smallest of the subsections because it is the farthest away from 3.5 to 5 meters.

As can be seen by our visual pathway diagram (figure 5.5.5) a distance away from the camera increases the size of the visual pathway decreases. The main reason for this is because the camera is placed about 8 inches above the ground. Thus, as objects get farther away from the camera, the number of pixels in the vertical direction decreases to travel the same amount of distance. Thus, the difference in the vertical bounds between the top and bottom of the green region is much smaller than the yellow or red but still of a similar change in distance from the camera. Additionally, we wanted to bring our users attention to objects closest to the camera rather than farthest; this is why we decided on the color green since it conventionally symbolizes to move forward and is the least cautionary of the colors utilized.

2.1.3. Subsystem 3: Power Supply Circuit

This subsystem is composed of a 12V, 5Ah sealed lead-acid (SLA) battery, two 10µF capacitors, and an LM1084 5.0 low dropout voltage regulator. These components are connected in parallel using 22 AWG single core wires with a diameter of approximately 0.64mm. These specific wires were used because their maximum current rating is 5A, and the maximum current drawn through this circuit is 4A [12]. The circuit schematic was created in MultiSim to test the efficiency of the voltage regulator and can be seen in Appendix figure 5.4.1. From the simulation, an input of 12V and 5A produced an output of approximately 5V and 5A. Our UP Board and its attached peripherals, including the GeeekPi7 monitor and the RealSense camera, only draws a maximum 5V of voltage and 4A of the current [12] so these simulation values were sufficient in concluding that the chosen components would be useful in supplying enough power to our device.

Some technical challenges arose when constructing this power supply circuit because the students quickly realized that SLA batteries are not supposed to draw current near their 1C rating (in our case, our 1C rating is 5A, which is only 1-1.5A higher than what we need for our device). In retrospect, had we used an SLA battery with a higher capacity rating, we might have been able to power our device for our intended period of use, 4 hours. This would require a 20Ah car battery, which is typically heavier than 10lbs in weight--this would have created a serious design flaw, as there is no place to put a battery this large on any mobility device. Because of this challenge, the students were unable to make our overall device portable or rechargeable, but we hope that another team can solve this issue with a different type of battery in the future that still complies with air travel regulations.

Besides our challenges, the students tested this power supply circuit for consumer safety by recording the temperature of the voltage regulator and its attached heat sink for 4 hours with thermistor probes connected to a multimeter to see if the heat sink used would effectively dissipate the heat acquired by the circuit. See Appendix figure 5.5.2 for the results of this test.

2.1.4. Subsystem 4: Encasements

An Intel Realsense D435 camera and Up Board cage for Permobil power wheelchair users is described. There are no current options to encase these devices. The optimal solution requires a method optimized for power wheelchair users. To address this need, a portable and adjustable cage was constructed using Solidworks 3D CAD 2017 software. Here, we will indicate the build features of the device; its ability to encase two different devices; and a method to attach the cage to a wheelchair.

The intended use of this embodiment allows the user to observe the environment behind them. The end user will be using a prototype encasement made of ABS material. The encasement for the present invention allows the user to install an Intel RealSense D435 camera and Up Board to the rear covering of his/her Permobil power wheelchair. These electronic devices can be disposed inside the main body. In identifying a method to secure the camera to the cage, the L-bar was designed. This design secures the camera to the cage via an L-bar, 3D-printed component made of ABS material, which has four screw holes (see Appendix figure 5.4.1 and figure 5.4.2.e). The top two holes secure the camera to the L-bar using two screws. The encasement has two holes which align to the bottom holes of L-bar. Once the camera is attached to the L-Bar, the L-bar can be secured to the encasement using to bolts. The top part of the L-bar is angled at 3 degrees to position the visual pathway with respect to the ground.

This encasement houses connective cords and wires, which run along the interior walls and connect the camera to the Up Board. An outlet on the rear of the encasement allows connective cords to attach to peripheral devices such as a 7 Inch 1024x600 Capacitive Touch Screen HDMI Monitor and an SLA battery (see Appendix figure.5.4.3.a). The UP Board can be placed freely inside the encasement. The back panel encloses the components described above via four holes for bolts to be placed (see Appendix figure.5.4.2.c and figure 5.4.4). Collectively, these components and body make up the weight of the encasement, which is 787 grams.

The rear cover for Permobil power wheelchairs is used to position the encasement (see figure 4.2.2.4). The encasement and rear cover have four hole openings which align to insert ¼-20 bolts (see Appendix figure.5.4.2.b). The bolts are required to attach the encasement to the rear cover 24/7. Additionally, the encasement can be separated such that it may be removed and/or replaced if necessary.

In an embodiment, for the monitor encasement is described. The encasement was designed using SolidWorks 2017 CAD software and can enclose the monitor and be mounted. Two opening features were designed to input an HDMI cord and USB import cord. The monitor encasement is comprised of two pieces, a bottom, and a top piece. The bottom of the encasement has a ¾ inch diameter holster, which is design to be inserted into a mic clip stand (see Appendix figure 5.4.5). The top of the encasement has an opening to allow the user to see the monitor display and can snap onto the bottom encasement.

2.2. Technical Challenges

Many technical challenges that were anticipated and encountered in the early stages of the project regarded the aspects of the device that every detail would be based on. Determining the components to use before building the device was a lengthy process, such as what to use to detect objects (IR Sensors, a camera, both), what microprocessor to use (Raspberry Pi, UP Board), coding language (Matlab, C++, Python), and operating system (Windows, Linux, Raspbian) were just some of the decisions that were made and remade at the beginning of the process. Once these were decided on, other challenges included figuring out how to modify the code from the given repositories that were made for our selected camera and implement them in a way that would meet our user stories.

Another technical problem encountered was producing a rechargeable, portable power supply for the device. The students attempted to use an SLA battery to supply the voltage and current needed by the UP Board and its attached peripherals. However, lead-acid batteries are known for being reliable for systems that draw low amounts of current--with our system drawing a maximum of 4A, the battery chemistry of an SLA battery with a maximum capacity of 12V/5Ah would never suffice. Even though our battery was capable of supplying the 4A of current we needed, this amount of current draw would only last approximately 70 minutes. If we incorporated a more powerful battery with a larger current capacity and perhaps a switching power supply circuit instead of a linear, voltage-regulated circuit, we might have seen better results. Because of these technical challenges, our minimal viable product is neither rechargeable nor portable because it still relies on a wall adapter power supply.

One of the significant technical challenges arose during the design process of the visual pathway. To determine distance with the visual pathway, the team thought to use Rs-Distance [21], a code from the SDK. Unfortunately, the code only calculates the distance of an object in the center of the screen. This is impractical because objects will not always be in an ideal position during use. The distance - from the camera - of pixels in other areas of the screen are necessary for optimal results. To fix this deficiency, the group attempted to utilize optical flow as a technique for pixel depth analysis. Unfortunately, this led to further issues such as latency. Low latency is essential for the device since a high latency can lead to safety issues for the user. The device latency was supposed to be lower than 300 milliseconds because the power wheelchairs maximum reverse speed is 5 mph and a latency greater than 300 milliseconds would leave the possibility that objects would approach from the rear on the screen before the user could react to them. Using optical flow, the latency was much higher. Therefore, to solve the visual pathway issues, the team decided to color individual pixels(table 5.5.2a), thus providing a structure for the user to assess object distance and minimizing latency.

2.3. System Testing

2.3.1 Integrated System Testing

To assure that the device would properly function when used by our client, the team needed to test all the subsystems together. When putting them together, we had a variety of concerns. The primary concern was if the camera was set at the right height. This was extremely important because if not oriented correctly, the visual pathway would not be displayed correctly. This brought up our secondary concern, the position of the camera. When bolted to the L-bar inside the encasement, the camera was placed at an angle of three degrees away from parallel to the encasements back edge; we were concerned that the camera might have obstructed the view. Additionally, the team had to ensure the display rested at viewable height and distance from the user. As for the encasement, the concern lay with it overheating and the security of the up board and camera fixation. To assure that all these concerns were dealt with, the team built the fully integrated system on the user's chair while the user sat in the chair. The encasement with the camera in it was attached to the back of the chair. Then the encasement with the monitor display was put into the holster near the armrest. Finally, the device was plugged into an outlet to provide the necessary power. Distance in intervals of 50 cms was marked behind the user's power chair. At each marked distance, an object was placed to verify it lay in the correct region on the visual pathway. Furthermore, once plugged in, the team monitored the heat of the encasement over the testing period to ensure it never became too hot to touch.

2.3.2: Testing RealSense-to-UP Board Subsystem

This subsystem was tested to ensure processing was efficient enough for use in a final device. The two variables tested for were the framerate of the data entering the UP Board and the time taken to initialize this process. Methods to test these variables involved adding print statements and timestamps after certain lines of code to find the difference in time between two different actions performed in the code. Testing requirements based on data sheets from the RealSense [8] and UP Board [9] include RealSense operating temperature cannot exceed 35°C and the UP Board Operating temperatures must remain within 0-60°C. Additionally, RealSense testing must fulfill standard 60950 regarding the safety of Information Technology Equipment [16] and standard JESD625B regarding devices that are electrostatic discharge sensitive [17].

An excerpt of code for unit testing is included in Appendix figure 5.3.5. Based on known framerate and initiation time from the data sheets and testing through a laptop (as opposed to the UP Board), a one-tailed t-test for each variable was performed in accordance with the null hypotheses below. Two tests were done for framerate, one for rs-capture.cpp and one for rs-imshow.cpp. This is due to the former file being much simpler than the latter, to test if there are significant differences in incoming framerate from the camera to UP Board, as shown in appendix table 5.5.7.

2.3.3: Testing the Monitor Display and Visual Pathway

Testing the monitor display was essential due to the importance of viewing the background effectively. The function related to this subsystem was data transfer and display. The specification being tested was a pixel density of 200 pixels per inch (PPI) +/- 100 pixels between 40 to 50 centimeters away. The PPI was calculated using the specifications of the monitor screen. In terms of visibility of the display from 40 to 50 centimeters away, a survey was conducted with participants found around the George Washington University's engineering hall. A total of 30 participants were surveyed at the distances of 40 and 50 centimeters away from the display. The display was hooked up to the camera and Up Board. Then, the participants were asked if they could see the display and its contents from those two distances. Following the survey, a z-test was conducted on the survey data from 50 centimeters away as every participant said yes to seeing the display from 40 centimeters away. This z-test data can be seen in Appendix table 5.5.6. The test experiment was aiming to answer the question of is the probability(P) of seeing the display from 50 centimeters greater than or less than 95%, refer to figure

To assure that the monitor display and visual pathway were statistically accurate and latency was within our specifications two tests were completed. The accuracy was measured and recorded in Appendix table 5.5.1. To acquire accuracy data the camera was placed at 12 inches above the ground at an angle of 3 degrees and the visual pathway was measured to see if it aligned with preset distances on the ground, a single proportion inference test was done to measure statistical significance. To measure latency and it's changing before and after implementation of the new visual pathway. The camera was pointed at a stopwatch and the time is shown on the stopwatch, and the monitor display was recorded twenty times resulting in an average latency of 311ms Statistical significance was recorded using a one-tailed t-test as displayed in Table 5.5.1b.

2.3.4: Testing Power Supply

The students observed the heat dissipated by the power supply circuit over the course of a 4 hour time period, as well as the output voltage and current of the circuit when connected to a 12V, 5Ah rechargeable SLA battery. According to the specifications of the voltage regulator we used, LM1084, our circuit junction temperature should not exceed 125°C for an extended period of time--to counteract the heat dissipated by this component, the students attached a large heat sink and recorded the temperature across it using thermistor probes. To measure the output voltage, the students attached a power resistor at the output of the circuit, rated at 25W and 1Ω, and used a voltmeter at each end of the resistor. This voltage was recorded over the course of 4 hours as well. To measure the output current of the power circuit, the power resistor was disconnected from the ground to break the circuit, and the ammeter probes were placed in series with the circuit with one probe at the end of the power resistor and the other attached to the ground. These three measurements were taken because output power and heat dissipation, when unregulated, can become dangerous to the consumer in high quantities.

2.3.5: Testing Encasements

This section specifies the material properties of ABS and the factor of safety for bolts made of stainless steel and how the material will supply the necessary tensile strength and elongation at break to holster the encasement and its components with a bolt factor of safety greater than or equal to two. An ABS material data sheet was required to determine tensile strength and elongation at break for the ABS material. Numerous simulation tests are offered in SolidWorks, such as static or drop test. The total weight of the encasement with its components was needed to determine the force placed on the bolts. The SolidWorks encasement and ¼-20 bolts were implemented and mated in assembly, and the simulation feature was selected. The ABS material and Stainless Steel (ferritic) was applied to the encasement and bolts, respectively. The Mass Properties feature gave an approximate weight of the encasement. The Ohaus Ranger 3000 determined the weight of the encasement and its components. The following formulas were used to determine max pressure on the surface of the encasement:

$$F=ma \qquad \text{(Equation 2.3.5.1)}$$

(F) stands for force measured in newtons. (m) is the mass of the encasement measured in kilograms. (a) is the acceleration due to gravity, m/s^2. The study chosen was a static simulation study. The max force applied on the bolts is equal to the mass of the encasement, converted to kilograms, and multiplied by 9.81 m/s^2. The max force is 7.7 N. Using the External Loads feature, 7.7 N was applied at the top of the encasement. Using the Fixtures feature, each of the four bolts was fixed. The research ran the study to determine the simulation results. Stress1 (-vonMises-), Strain1(-Equivalent-), and Factor of Safety (-FOS-) was used to determine the tensile strength of the ABS, elongation at break of the ABS material, and the safety factor of the bolt, respectively. Simulations results were then compared to the ABS material data sheets and the safety factor requirement (see Appendix table 5.5.4).

H_1 = The ABS tensile strength and elongation at break will holster the encasement and its components with a bolt factor of safety greater than or equal to two.

H_0 = The simulation has a tensile strength and elongation higher than the ABS material data sheet with a bolt factor less than two, and thus the encasement will not hold the weight of its components when connected to the power wheelchair.

3. Results

3.1. Test Results

Our main testing results regard the overall system accuracy when in use. Others regard latency and accuracy of the camera and visual pathway, operation of the power supply, and the safety and efficacy of encasements and mounts.

3.1.1 Integration Testing Results (include display/visual pathway results here)

Using an online PPI calculator, the screen's diagonal length and resolution were inputted. The calculator showed the PPI of the screen to be 143.5, which met specifications. Furthermore, the results in Appendix table 5.5.5 and 5.5.6 show that 29 people out of 30 stated they could see the display from 50 cms away. A z-test was conducted in order to determine statistical significance. The null hypothesis stated that the probability of viewing the display was less than 95%. The z-test rejected the null hypothesis, showing the team that the display met the specification of visibility from 50 cms away.

Data analysis (using an inference for single-proportion) for accuracy of the visual pathway showed that the accuracy of the pathway met specifications, that on average it would be 95% correct with a confidence of 95%. The results are shown, in Appendix table 5.5.1, for each part of the visual pathway. Moreover, data analysis using a one-tailed T-test showed that there was only a 3.3% percent chance our data would meet our specification of 300ms; thus we failed to reject our null hypothesis that our latency was above 300ms. Data was recorded twenty times and is shown in Appendix table 5.5.2a.

3.1.2 Camera-to-UP Board Results

The results in Appendix figure 5.5.1 and table 5.5.3 show statistical significance that the UP Board is not as fast at processing as a laptop is when it comes to running executable files. However, this evidence is not practically significant. Initiation time is still under a second for the UP Board to begin the process; there is little concern for the user under the assumption that he waits for the display to begin moving backward while using the system. The frame rate, while approximately 2/3 of the maximum framerate from the RealSense datasheet, is still transferring ~19 fps. When these frames are displayed (which is implemented later), the human eye has difficulty noticing such a small difference.

3.1.3 Power Supply Results

The power supply circuit failed to output the current necessary to power the UP Board and its peripheral attachments. The initial output of current was measured across a 1Ω, 25W power resistor to be 3.7A, but this current quickly dissipated to less than 1A in under a minute's time. The output voltage remained at approximately 5V over time until the battery began to lose its charge, which was observable after approximately 2 hours. The temperature of the voltage regulator and its attached heat sink remained under the maximum circuit temperature, 125°C, for most of the 4 hours during which testing was conducted. Toward the 3rd hour, however, the heat sink temperature rose to just over the limit, eventually plateauing at approximately 127°C. The graphical representation of the temperature data can be found in Appendix figure 5.5.2.

3.1.4 Encasement Results

The material justification for the prototype encasement made of ABS material passes. The simulation tensile strength is 4.756e-002 N/m^2, as shown in Appendix table 5.5.4. The simulation elongation at break is 2.784e-006. The simulation factor of safety is 1.1e+003. The simulation results for tensile strength and elongation at break are lower than the ABS material sheet justification. The factor of safety for the ¼-20 bolts is great than 2. The weight of the encasement will uphold on the back of a power wheelchair. The test rejects the null hypothesis and accepts the alternative hypothesis.

3.2. Accomplishments.

Our general goal for a working code with all components and encasements was a success. Our device can successfully power up with two codes that display the environment behind the user, one with and one without a visual pathway. The camera and UP Board are contained in an encasement that attaches to the back of the client's chair, and the monitor is within an encasement attached to the arm of the user's chair that can be adjusted for preferable viewing. We met the following user stories:

- The user will be able to view the environment behind them.
- The user will be able to gauge the depths of objects behind them.
- The user will be able to adjust the display for a preferred viewing angle.

3.3. Setbacks

The portable power system never worked perfectly for our device. While we were getting the amount of necessary voltage, our circuit never produced enough current to power the system. The alarm system was also never accomplished. The auditory cue code, even when tested through a computer with a larger processor, never worked with the accuracy and latency period necessary for it to be included in the final product. Due to this problem, a speaker circuit was never built for the system as there was no working code nor a built-in speaker on the monitor that could be utilized to implement it.

3.4. Discussion.

All aspects of the backup detection device should be taken into consideration and have general design ideas before it becomes too late in the semester to put everything together. Specifications such as required current, voltage, or power are helpful to know off-hand to prevent a rush of trying to power a device with little time left in the project. More research should be done for designing the project rather than blindly attempting to build subsystems.

3.5. Recommendations & Future Work.

The portable power system would have to be modified to produce the required current for the UP Board to make this a fully useable product. Ideally, this device should be able to detect a more exact distance of objects in the user's path and not just give them a general idea of where the objects are. There are many other features this system would benefit from that add value and utility to the final product. A visual pathway that curves with the direction of the chair as it backs up, a working audio system, and night vision would be useful to have to help the user to make the device more adaptable in different situations. The current encasement for the user is a prototype made of ABS material. An injection mold, 3D printed proxy, made of PC+ABS would be an ideal material for future replication of the device. A PC+ABS blend has higher tensile strength and elongation compared to ABS alone. The test described for the current encasement in this document will be conducted to study the PC+ABS blend.

4. User Interactions/Manual

4.1 Description

This device works to detect objects behind any operational power chair with visual warnings to the user for different scenarios, such as objects moving into their path or still objects they would not be able to see without a camera. It observes, analyzes, and displays the environment behind the user, detects the objects, and alerts the user. The device is compatible with all powerchairs, easy to set up, and has reliable and safe components such as a power pack, adjustable mounts, and a clean user interface.

4.2 Assembly

4.2.1 Components

Components included in this backup detection system can be seen in Figure 4.1. These include an Intel RealSense D435 Depth Camera, a UP Board processor, GeeekPi 7 Inch Touch Screen LCD Monitor, and a rechargeable, portable power supply. Cables include a USB-3 to USB-C for connecting the camera to the UP Board, an HDMI and USB to micro-USB to connect the UP Board to the monitor, and a 5V @ 4A power jack to power the UP Board and cables to recharge the battery. Additionally, an encasement to hold the power bank, UP Board, and camera and an adjustable monitor mount are also included. The encasement components include one L-Bar, one back panel, two silver M3 screws, two black M3 screws, washers, and nuts, four ¼ -20-inch bolts and nuts, and four M4 bolts and nuts.

4.2.2 General Assembly

Intel Realsense D435 camera assembly and attachment:

An Intel Realsense D435 camera and Up Board encasement for Permobil power wheelchair user set up is described. To assemble the system, first, connect the camera to the L-Bar. Place the Intel RealSense D435 USB-3 to USB-C cable into the camera import. Align the back of the L-Bar to the back of the camera. The screw holes from the camera and L-Bar should match. Screw two silver M3 screws into the top holes of the L-Bar until the camera is secured (see figure 4.2.2.1.c). At this point, the camera USB-3 to USB-C cable should be attached to the camera and the L-bar to the camera. Next, place the camera and L-Bar into the encasement and align the bottom holes of the L-Bar to the two holes placed at the bottom of the encasement. Place a washer on both of the black M3 screws. Place two black M3 screws from the outside of the encasement into the L-Bar (see figure 4.2.2.1.d). Using M3 nuts, tighten the black M3 screws until the L-Bar is secured to the encasement. Now the camera is secured into the encasement (see figure 4.2.2.1), if not repeat the steps above.

UP Board assembly and placement:

Second, the Up Board placement into the encasement is described. Insert the monitor USB, HDMI and the 5V @ 4A power jack into the encasement outlet. Insert the monitor USB into the Up Board USB import (see figure 4.2.2.2.e) Ensure that the USB is connected to the "USB" input and not "Power." To power on the system, plug the 5V power jack into the UP Board(see figure 4.2.2.2.c). To display the background environment onto the monitor, place the HDMI into the designated UP Board import(see figure 4.2.2.2.b). Once completed, insert the UP Board into the encasement. For additional assistance, see figure 4.2.2.2, for a descriptive image of the cable connections and UP Board placement.

Back panel and system assembly and attachment to the chair:

Third, place the back panel onto the back of the encasement and place four silver M4 screws and nuts into the appropriate holes (see figure 4.2.2.3). Fourth, the encasement attachment to the chair back panel is described. Ensure the chair back panel is not connected to the chair. Place four ¼ -20-inch bolts holes from the front of the encasement into the designated holes. Place the front of the chair pack panel onto the back of the encasement. Secure the encasement onto the chair back panel via four ¼ -20-inch nuts. See figure 4.2.2.4 for additional support. Once the encasement is secured to the back panel, attach the system to the chair using the designed back panel attachments.

Monitor and battery assembly and attachment:

Fifth, the monitor to encasement connection set-up is described. The monitor encasement is comprised of two components. The first component (bottom case) has a handle and the other (top case) secures the monitor. Place the monitor into the component with the handle. Place the top case over the bottom case and monitor. Ensure the monitor outlets align with each of the cases outlets. The monitor is now secure in its case. Place the handle inside the mic-clip. Connect the HDMI and USB to micro-USB to connection to the monitor. Lastly, connect and the place the charged battery into the backpack on your back upholstery. Your system is now fully completed and ready to be used.

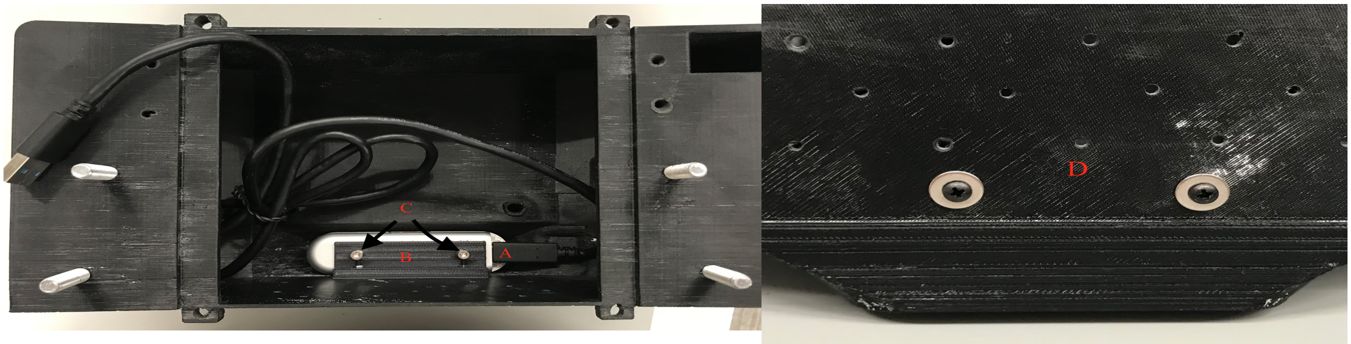


Figure 4.2.2.1 *Intel Realsense D435 camera assembly and attachment. (**a**) USB-3 input to the camera. (b) the L-Bar attachment. (**c**) Two silver M3 screw placement to secure the camera to the L-bar. (**d**) The outer encasement placement for two black M3 screws to secure the L-Bar.*

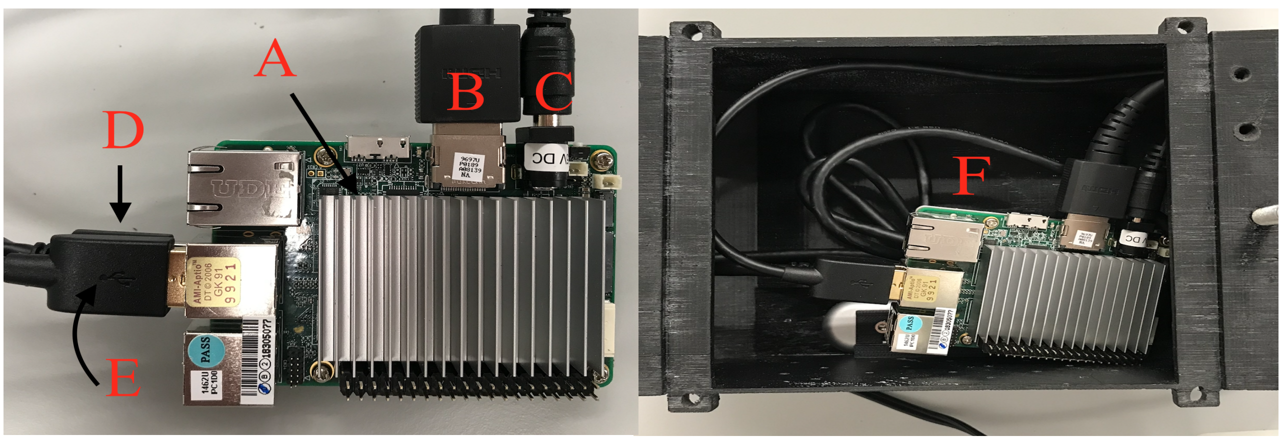


Figure 4.2.2.2 *UP Board assembly and placement. (**a**) The Up Board Microcontroller. (**b**) The HDMI connection. (**c**)The 5V @ 4A power jack connection. (**d**) The USB-C connection for the camera. (**e**) The USB connection for the monitor. (**f**) Up Board placement.*

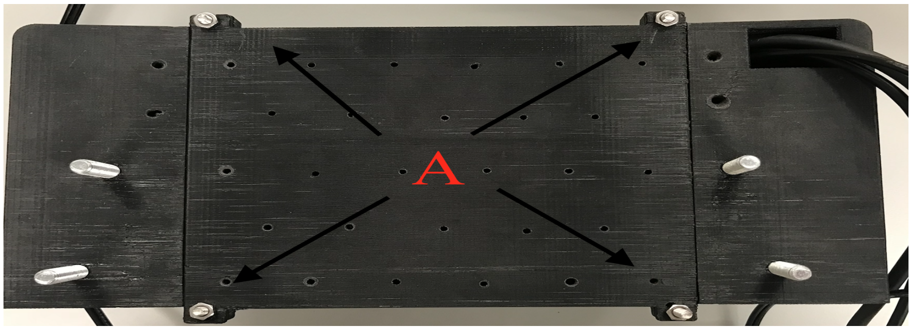
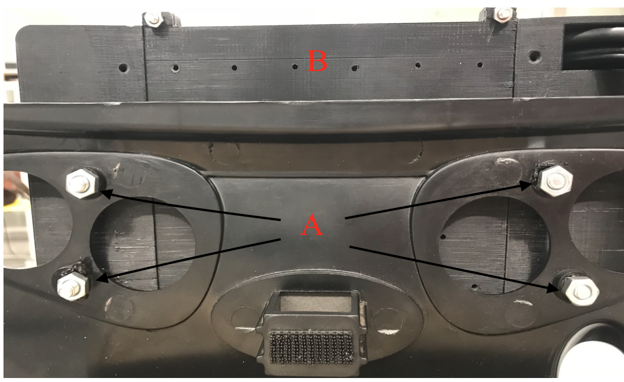


Figure 4.2.2.3. *Back panel placement. (**a**) placement for M4 screws. Secure the bolts using the nuts. This will ensure that the back panel is secured to the encasement.*



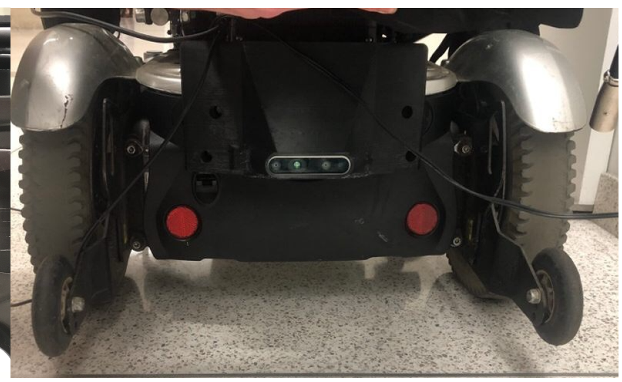


Figure 4.2.2.4. *Encasement attachment. (**a**) placement for four ¼-20-inch bolts and nuts. (**b**) Position for the encasement.*

4.2.3 Recharging the Power Supply

To recharge the power supply, attach the two leads from the 12V trickle charger onto the battery at the matching colored nodes, red lead to the red terminal and black lead to the black terminal. A red light will show on the trickle charger while the battery is charging, and a green light will show when it is fully charged.

4.3 User Interaction—General Interface

Once components are assembled, test the live stream abilities of the device. Ensure the power bank and monitor are on and double tap the "LIVESTREAM" or "PATHWAY" applications on the home screen. The camera should immediately record live video and display it on the monitor and should appear like one of the figures below with the correct background behind the power chair.

For LIVESTREAM a simple visual of the environment should show. For PATHWAY, the system will create a visual pathway to give an idea of where an object is. This angle of the camera within the encasement may need to be modified to calibrate the distances correctly. Place a still object at a known distance and back up towards it, ensuring the colored zone and audible alerts are in accordance with table 4.3.1. Once correct, the encasement can be properly screwed into the back of the powerchair.

Zone	Distance (Meters)	Color in Pathway
Safe	5+	N/A
Far in Path	3.5-5.0	Green
Close in Path	1.5-3.5	Yellow
Very Close in Path	0.75-1.5	Red
Dangerously Close in Path	0.0-0.75	Red

Table 4.3.1: Visual pathway depth threshold guidelines

4.4 User Interaction—Modify Interface

To manipulate what is seen and heard from the available systems, buttons on the interface will adjust the alarms based on user preferences, seen in Figure 4. On the left, the "Visual Pathway" button with the eye icon will show or hide the appearance of the visual pathway when pressed. The colored lines will no longer appear on the interface. The center button with the bell icon turns off the alarm system when pressed. This means the audible alarm and visual red overlays will not sound or appear when the system detects objects. The button on the far right will mute the audio system when pressed. Audible beeps for still or moving obstacle detection will not sound, but visuals will still appear.

The user will also be able to adjust general monitor settings such as brightness and volume of the display. The power source should be placed in a location accessible enough so the user can turn it on and off when it is or is not in use.

4.5 Troubleshooting

Problems encountered can include no display when powered on, or no video stream. When turning on or setting up the detection system and facing these issues, check the following components to ensure they are properly connected and turned on.

- UP Board—Camera
- UP Board —Power
- UP Board—Monitor
- Power bank is charged

5. Appendices

5.1. Objectives, Functions, and Specifications

Objectives & Specifications

- Device must have a visual pathway to guide user.
  - Visual pathway must accurately reflect distance thresholds.
- Device must have an alarm system for approaching obstacles.
  - Volume should be preference of the user [5].
- Monitor must be adjustable and supported.
  - Mount with arm that can be rotated and support the weight of monitor with 150% safety margin.
- Power source must be portable, rechargeable, and protected from the outside environment.
  - Battery capacity < 100Wh [6].
  - Non-spillable, dry or gel battery [6].
  - Battery life >= 4 hours of continuous use.
  - Charging time < 12 hours (overnight) [7].

Functions & Specifications

- Image acquisition (observe the environment behind the chair).
  - Pixel density of at least 200ppi +/- 100ppi for viewing distance of 40-50cm [4].
- Data transfer and display (display the environment to the user).
  - Avg. camera-to-display latency < 300ms.
- Measure the distance of obstacle to chair with high accuracy.
  - Distance must be measured over 95% accuracy (actual vs calculated distance).
- Alert the user of obstacles via audio cue (beep).
  - Sound should be between 150-1000Hz, with four harmonics 300 to 4000 Hz each [5].

The alarm system was never incorporated into the project because it produced a significant amount of latency in the display of objects on the monitor. Therefore, objective 2 and function 4 were never completed. Additionally, the choice of the battery was inadequate to supply the power the system needed for 4 hours, so a wall adapter was used in the final demo to test the integrated system.

5.2. Standards

- *"*Information Technology Equipment – Safety – Part 1: General Requirements*"* UL Standard 60950-1, 2007.
- "Requirements for Handling Electrostatic-Discharge-Sensitive (ESDS) Devices" JEDEC Standard JESD625B, 2011
- "IEEE Standard on Video Techniques: Measurement of Resolution of Camera Systems" *1993 Techniques,* IEEE Standard 208, 1995. ISO 2580-1:2002
- "Federal Motor Vehicle Safety Standard, Rearview Mirrors; Federal Motor Vehicle Safety Standard, Low-Speed Vehicles Phase-In Reporting Requirements," *Federal Register*, 07-Dec-2010. [Online]

- "Batteries Carried By Airline Passengers: Frequently Asked Questions." Federal Aviation Administration Office of Hazardous Materials Safety. Based on US DOT regulations (49 CFR, Sec. 175.10). Revised Sep. 9, 2016.
- *Information Technology Equipment – Safety – Part 1: General Requirements,* UL Standard 60950-1, 2007.
- *Requirements for Handling Electrostatic-Discharge-Sensitive (ESDS) Devices,* JEDEC Standard JESD625B, 2011
- *IEEE Standard on Video Techniques: Measurement of Resolution of Camera Systems, 1993 Techniques,* IEEE Standard 208, 1995.

5.3. Code

```
// License: Apache 2.0. See LICENSE file in root directory.
// Copyright(c) 2017 Intel Corporation. All Rights Reserved.

#include <librealsense2/rs.hpp> // Include RealSense Cross Platform API
#include "example.hpp"          // Include short list of convenience functions for rendering

// Capture Example demonstrates how to
// capture depth and color video streams and render them to the screen
int main(int argc, char * argv[]) try
{

   rs2::log_to_console(RS2_LOG_SEVERITY_ERROR);
   // Create a window for rendering:
         window app(800, 600, "RealSense Capture Example");

   // Declare rates printer for showing streaming rates of the enabled streams.
   rs2::rates_printer printer;

   // Declare RealSense pipeline, encapsulating the actual device and sensors
   rs2::pipeline pipe;

   // Start streaming with default recommended configuration
   // The default video configuration contains Depth and Color streams
   // If a device is capable to stream IMU data, both Gyro and Accelerometer are enabled by
default
   pipe.start();

   while (app) // While the application is still running
   {

                  rs2::frameset data = pipe.wait_for_frames(). // Wait for next set of frames from
camera
                           apply_filter(printer);     // Print each enabled stream frame rate

```

```
                    app.show(data);

   }

   return EXIT_SUCCESS;
}
catch (const rs2::error & e) //Error functions if code is not functioning correctly
{
   std::cerr << "RealSense error calling " << e.get_failed_function() << "(" <<
e.get_failed_args() << "):\n    " << e.what() << std::endl;
   return EXIT_FAILURE;
}
catch (const std::exception& e)
{
   std::cerr << e.what() << std::endl;
   return EXIT_FAILURE;
}
```

Figure 5.3.1: rs-capture.cpp fully commented

```
// License: Apache 2.0. See LICENSE file in root directory.
// Copyright(c) 2017 Intel Corporation. All Rights Reserved.

#include <librealsense2/rs.hpp> // Include RealSense Cross Platform API
#include <opencv2/opencv.hpp>   // Include OpenCV API
#include <opencv2/dnn.hpp>
#include "../cv-helpers.hpp"
#include <windows.h>
#include<opencv2/core/core.hpp>
#include<opencv2/highgui/highgui.hpp>
#include<opencv2/imgproc/imgproc.hpp>
#include<iostream>

int main(int argc, char * argv[]) try
{
        // Declare depth colorizer for pretty visualization of depth data
        rs2::colorizer color_map;

        // Declare RealSense pipeline, encapsulating the actual device and sensors
        rs2::pipeline pipe;

        // Start streaming with default recommended configuration
        pipe.start();

        using namespace cv;
```

```
	using namespace cv::dnn;
	using namespace rs2;
	using namespace std;
	int16_t *val;					//a pointer to the data in a depth frame
	int w;
	w = 800; //Width of window for display
	int h;
	h = 600; //Height of window for display
	int16_t timg1[800][600];

	const auto window_name = "Display Image";
	namedWindow(window_name, (w, h)); //// Create a window for rendering

	while (waitKey(1) < 0 && cvGetWindowHandle(window_name))
	{

		rs2::frameset data = pipe.wait_for_frames(); // Wait for next set of frames from
camera
		auto color_frame = data.get_color_frame();//Acquire color frames from
frameset
		Mat color_mat = frame_to_mat(color_frame);//Convert type frame to type Mat
for edits		Mat video;
		resize(color_mat, video, Size(w, h), 0, 0); //Resize frame to fit in window

		Mat dst, pathway, overlay;

		cv::Mat final;
		cv::flip(video, final, 1); //Flip video to appear mirrored during use


		//Colorizing Pixels
		for (int i = 366; i < 435; i++)//Create top green line horizontal from x-pixels 366
to 434
		{
			for (int j = 300; j < 303; j++) //Line is 3 pixels tall from y-pixel 300 to 302
			{
				Vec3b color = final.at<Vec3b>(Point(i, j)); //Acquire color of
pixels

				{
	//Define color of pixels to RGB green
					color[0] = 0; //B
					color[1] = 255; //G
					color[2] = 0; //R
				}
				final.at<Vec3b>(Point(i, j)) = color; //Assign color to pixels
			}
```

```
                }


                for (int i = 345; i < 457; i++) //Create top yellow line horizontal from x-pixels
345 to 456
                {
                        for (int j = 316; j < 319; j++) //Line is 3 pixels tall from y-pixel 316 to 319
                        {
                                Vec3b color = final.at<Vec3b>(Point(i, j));//Acquire color of
pixels

                                {
              //Define color of pixels to RGB yellow
                                        color[0] = 0; //B
                                        color[1] = 255; //G
                                        color[2] = 255; //R
                                }
                                final.at<Vec3b>(Point(i, j)) = color; //Assign color to pixels
                        }
                }

                for (int i = 239; i < 564; i++) //Create top red line horizontal from x-pixels 239 to
563
                {
                        for (int j = 403; j < 406; j++) //Line is 3 pixels tall from y-pixel 403 to 405
                        {
                                Vec3b color = final.at<Vec3b>(Point(i, j)); //Acquire color of
pixels

                                {
              //Define color of pixels to RGB red
                                        color[0] = 0; //B
                                        color[1] = 0; //G
                                        color[2] = 255; //R
                                }
                                final.at<Vec3b>(Point(i, j)) = color; //Assign color to pixels
                        }
                }

                for (int i = 1; i < 799; i++) //Create bottom red line horizontal from x-pixels 1 to
798
                {
                        for (int j = 597; j < 600; j++) //Line is 3 pixels tall from y-pixel 403 to 599
                        {
                                Vec3b color = final.at<Vec3b>(Point(i, j)); //Acquire color of
pixels
```

```
				{
			//Define color of pixels to RGB red
					color[0] = 0; //B
					color[1] = 0; //G
					color[2] = 255; //R
				}

				final.at<Vec3b>(Point(i, j)) = color; //Assign color to pixels
			}
		}

		float m;
		int b = 599; //y-intercept of left horizontal
		m = -0.8156; //slope of left horizontal
		for (int i = 1; i < 366; i++) //Left horizontal from x-pixels 1 to 365
		{
			float j;
			j = (round(m * i + b)); //Determine j based on x, m, and b
			//Acquire color of pixels to create 3-pixel wide horizontal
	Vec3b color = final.at<Vec3b>(Point(i, j));
			Vec3b color2 = final.at<Vec3b>(Point(i, (j + 1)));
			Vec3b color3 = final.at<Vec3b>(Point(i, (j - 1)));

			if (i < 240) //Define color of x-pixels 1 to 239 to RGB red
			{
				color[0] = 0;
				color[1] = 0;
				color[2] = 255;

				color2[0] = 0;
				color2[1] = 0;
				color2[2] = 255;

				color3[0] = 0;
				color3[1] = 0;
				color3[2] = 255;
			}
			else if (i < 347 && i > 239) //Define color of x-pixels 240 to 346 to RGB
yellow
			{
				color[0] = 0;
				color[1] = 255;
				color[2] = 255;

				color2[0] = 0;
				color2[1] = 255;
				color2[2] = 255;
```

```
                    color3[0] = 0;
                    color3[1] = 255;
                    color3[2] = 255;
                }
                else if (i < 366 && i > 346) //Define color of x-pixels 347 to 365 to RGB
green
                {
                    color[0] = 0;
                    color[1] = 255;
                    color[2] = 0;

                    color2[0] = 0;
                    color2[1] = 255;
                    color2[2] = 0;

                    color3[0] = 0;
                    color3[1] = 255;
                    color3[2] = 0;
                }
    //Assign specified colors to pixels
                final.at<Vec3b>(Point(i, j)) = color;
                final.at<Vec3b>(Point(i, (j + 1))) = color2;
                final.at<Vec3b>(Point(i, (j - 1))) = color3;


            }
            float m1;
            int b1;
            b1 = -54;   //y-intercept of right horizontal
            m1 = 0.8156; //slope of right horizontal
            for (int i2 = 434; i2 < 800; i2++) //Right horizontal from x-pixels 434 to 799
            {
                float j2;
    j2 = (round(m1 * i2 + b1));//Determine j based on x, m, and b
    //Acquire color of pixels to create 3-pixel wide horizontal
                Vec3b color = final.at<Vec3b>(Point(i2, j2));
                Vec3b color2 = final.at<Vec3b>(Point(i2, (j2 - 1)));
                Vec3b color3 = final.at<Vec3b>(Point(i2, (j2 + 1)));

                if (i2 < 800 && i2 > 560) //Define color of x-pixels 559 to 799 to RGB
red
                {
                    color[0] = 0;
                    color[1] = 0;
                    color[2] = 255;
```

```
                    color2[0] = 0;
                    color2[1] = 0;
                    color2[2] = 255;

                    color3[0] = 0;
                    color3[1] = 0;
                    color3[2] = 255;

               }
               else if (i2 < 561 && i2 > 453) //Define color of x-pixels 454 to 560 to
RGB yellow
               {
                    color[0] = 0;
                    color[1] = 255;
                    color[2] = 255;

                    color2[0] = 0;
                    color2[1] = 255;
                    color2[2] = 255;

                    color3[0] = 0;
                    color3[1] = 255;
                    color3[2] = 255;
               }
               else if (i2 < 454 && i2 > 434) //Define color of x-pixels 435 to 453 to
RGB green
               {
                    color[0] = 0;
                    color[1] = 255;
                    color[2] = 0;

                    color2[0] = 0;
                    color2[1] = 255;
                    color2[2] = 0;

                    color3[0] = 0;
                    color3[1] = 255;
                    color3[2] = 0;
               }
     //Assign specified colors to pixels
               final.at<Vec3b>(Point(i2, j2)) = color;
               final.at<Vec3b>(Point(i2, (j2 - 1))) = color2;
               final.at<Vec3b>(Point(i2, (j2 + 1))) = color3;
          }

          imshow(window_name, final); //Display outut on monitor
```

```
        }

        return EXIT_SUCCESS;
}



catch (const rs2::error & e)
{
        std::cerr << "RealSense error calling " << e.get_failed_function() << "(" <<
e.get_failed_args() << "):\n    " << e.what() << std::endl;
        return EXIT_FAILURE;
}
catch (const std::exception& e)
{
        std::cerr << e.what() << std::endl;
        return EXIT_FAILURE;
}
```

Figure 5.3.2: rs-imshow.cpp fully commented

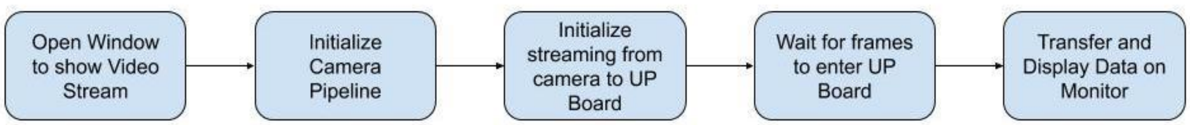


Figure 5.3.3: High-level algorithm of rs-capture.cpp

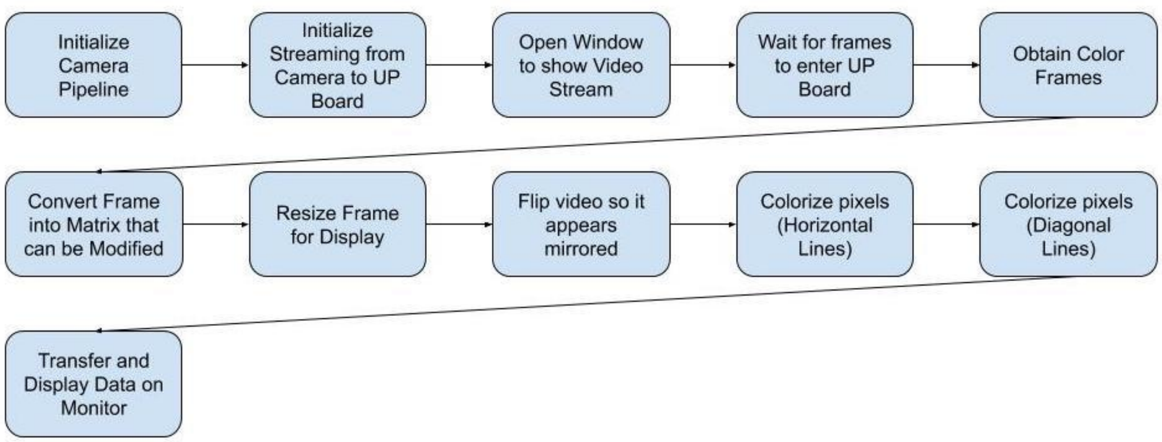


Figure 5.3.4: High-level algorithm of rs-imshow.cpp

```
typedef std::chrono::high_resolution_clock Clock;
int main(int argc, char * argv[]) try
{
    rs2::pipeline pipe;
    auto t1 = Clock::now(); //time of pipeline initiation
    pipe.start();
    auto t2 = Clock::now(); //timestamp of stream initiation

    std::cout << "Initiation time: "
    << std::chrono::duration_cast<std::chrono::milliseconds>(t2 - t1).count()
    << " milliseconds" << std::endl; //prints initiation time (ms)
    const auto window_name = "Display Image";
    while (waitKey(1) < 0 && cvGetWindowHandle(window_name))
    {
        rs2::frameset data = pipe.wait_for_frames(); // Wait for next set of frames from the camera
        auto t3 = Clock::now();
        std::cout << std::chrono::duration_cast<std::chrono::milliseconds>(t3 - t1).count()
        << std::endl;
```

Figure 5.3.5: Excerpt of rs-imshow.cpp for testing camera-to-UP Board Subsystem

5.4. Engineering Drawings

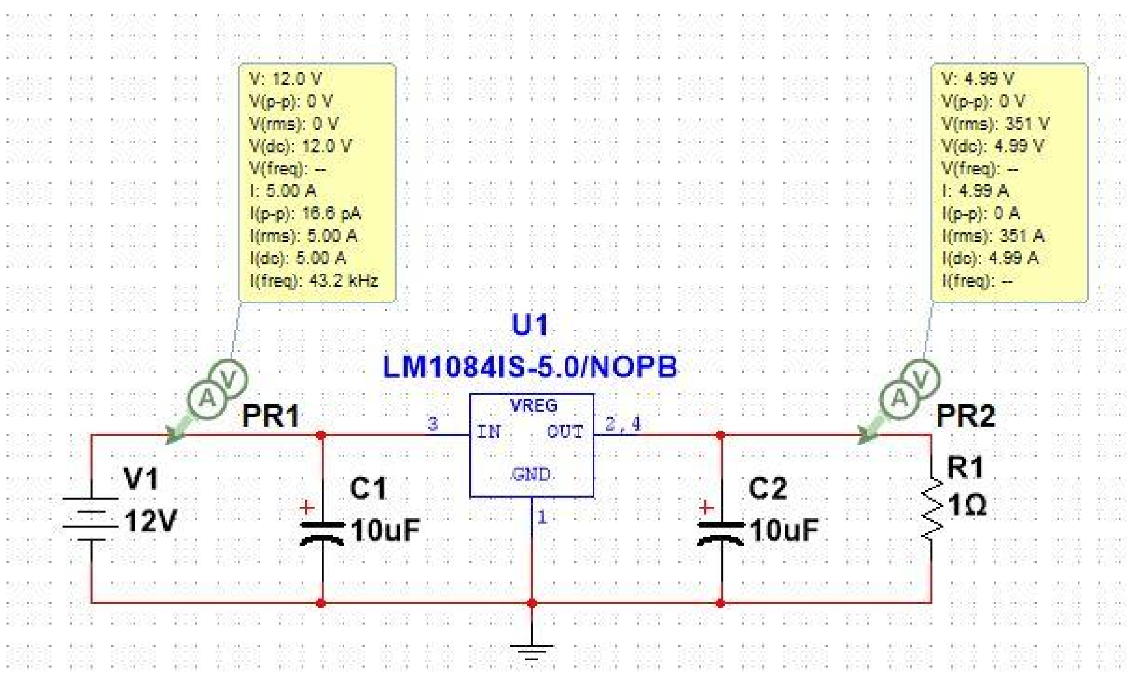


Figure 5.4.1. *The MultiSim circuit schematic of the power circuit which converts 12V and 5A of input power into 5V and 5A of output power. Two 10µF capacitors were used in parallel at the input and output nodes of the LM1084 voltage regulator in order to reduce noise and protect the*

regulator and the load from any surge of current. 22 AWG single core wires were used to connect the components of the circuit because they have a maximum current rating of 5A [12].

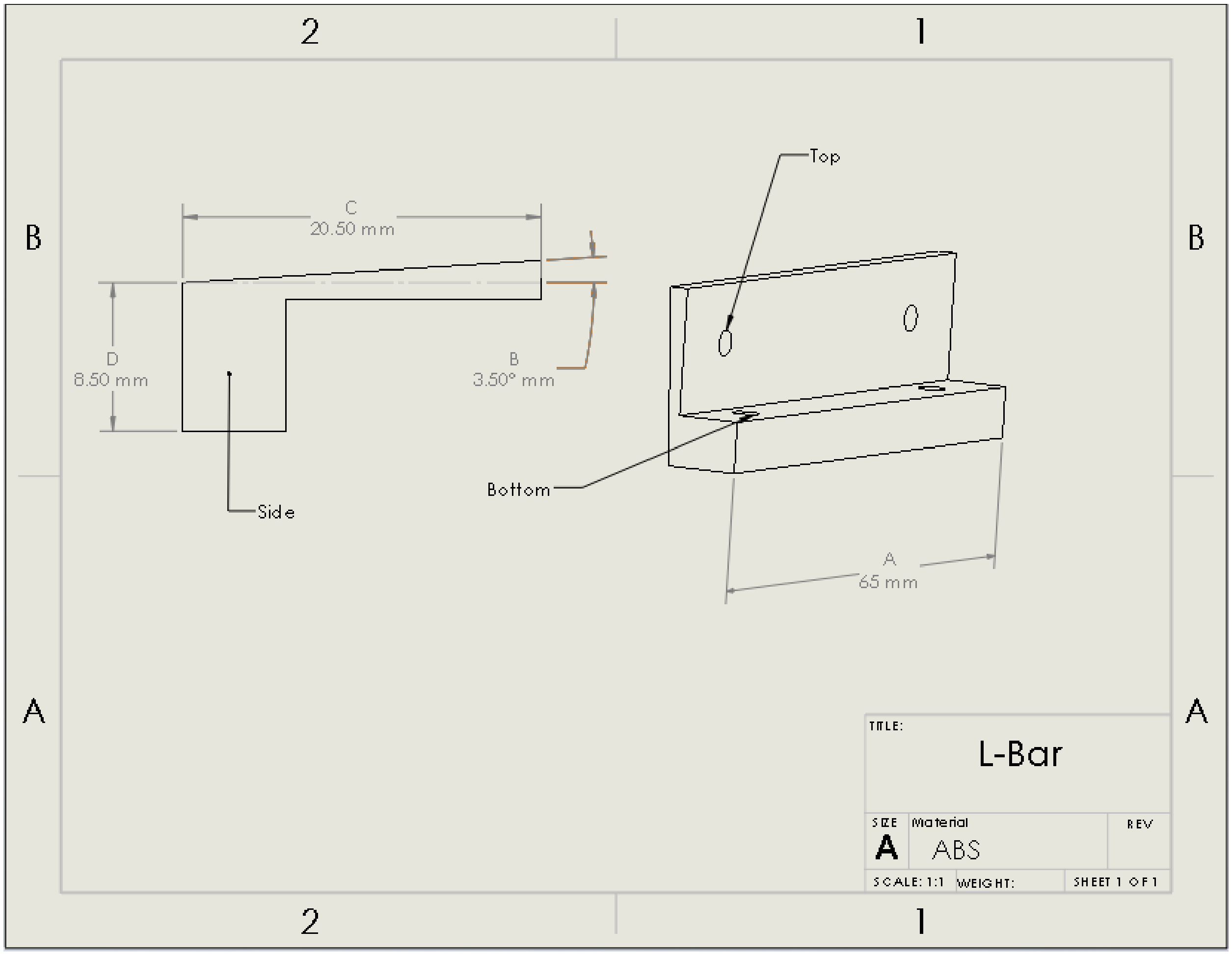


Figure 5.4.2. *The L-bar CAD design for the encasement. (**a** and **d** and **c**) Length, width, and height of the L-Bar. The camera will attach to the back of the topside. M3 Screws will be placed at the top and screwed into L-Bar and camera. The bottom holes are used to screw the L-Bar to the encasement via M3 bolts. (**b**) This angle positions the camera 86.5 degrees above the ground. The L-Bar material is ABS. The weight of the L--Bar is 5.02 grams.*

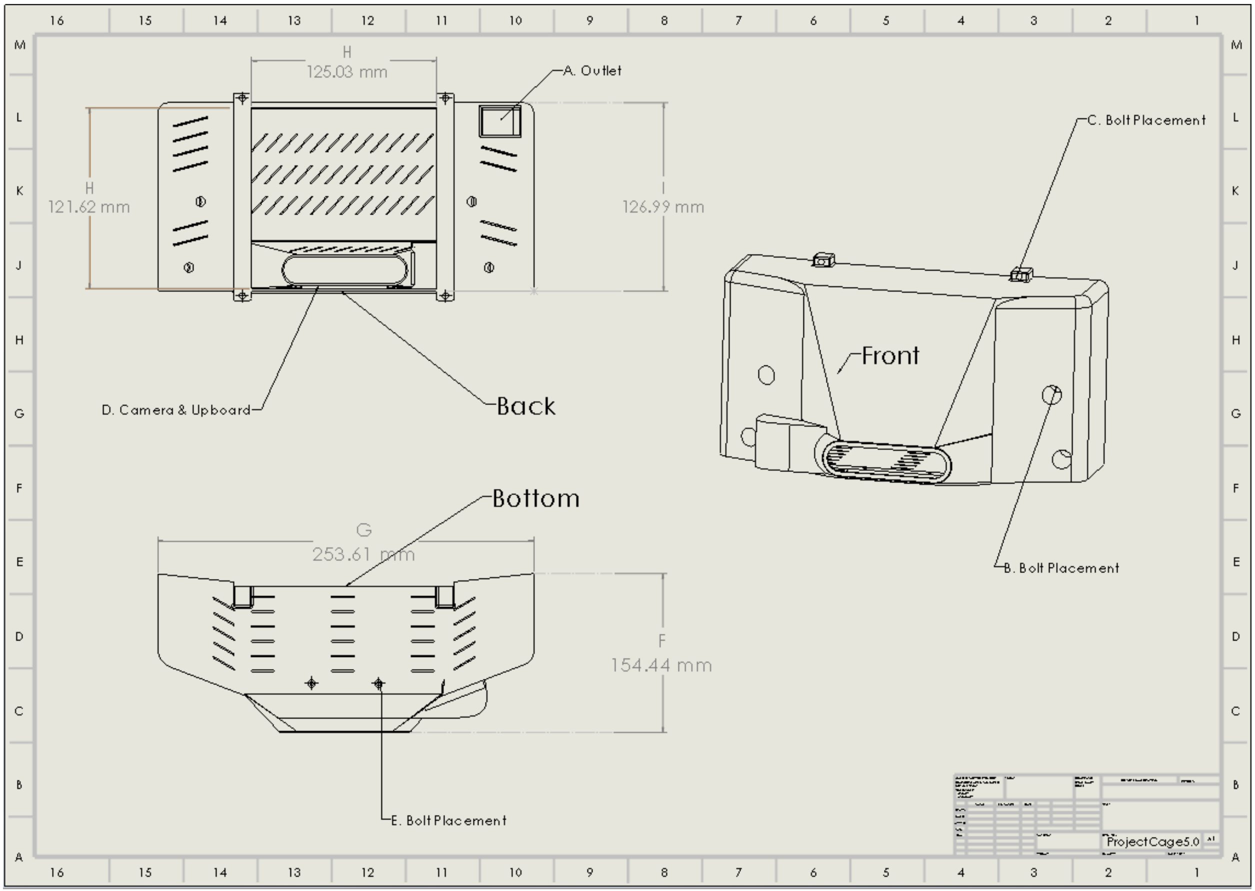


Figure 5.4.3. *A CAD drawing of the encasement. (**a**) The outlet for the battery power cord, HDMI cord, and monitor power cord. (**b**) Bolt placement for M6 bolts. Four bolts are required to place encasement onto the user's chair panel. (**c**) Bolt placement for M4 bolts. Four bolts are required to seal the back panel to the encasement. (**d**) Camera and Up Board placement. (**e**) Bolt placement for them to seal the L-Bar to the encasement. (**f**) the Width of the encasement. (**g**) The length of the encasement. (**h**) Dimensions to required to fit the Up Board. (**I**) The height of the encasement. Encasement material is ABS. Encasement weight is 466.67 grams.*

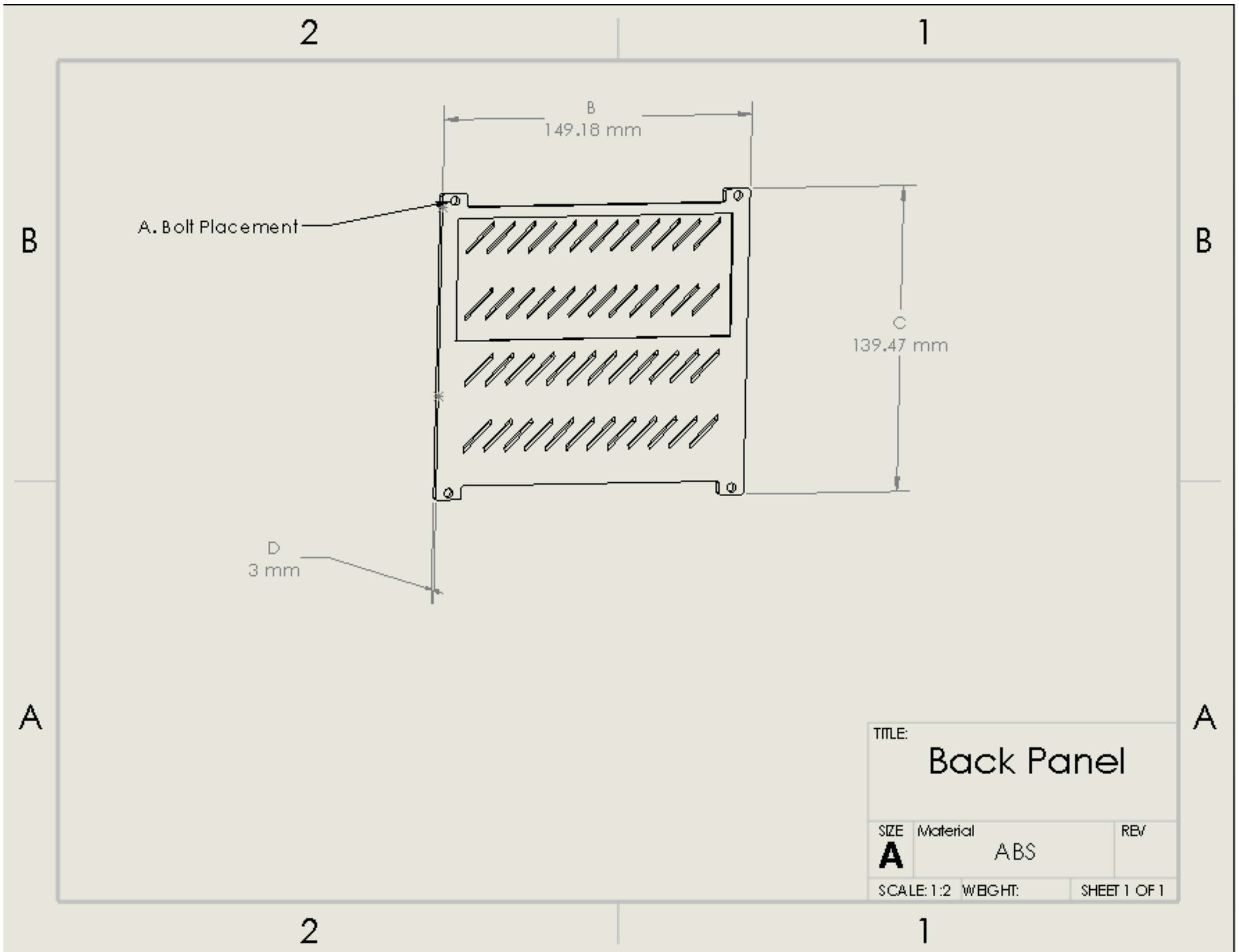


Figure 5.4.4. *A CAD drawing of the back panel. (**a**). Bolt place to seal the panel to the encasement and enclose all components. (**b** and **c**) The length and width of the back panel. (**d**) The thickness of the back panel. The back panel material is ABS. Back panel weight is 44.13 grams.*

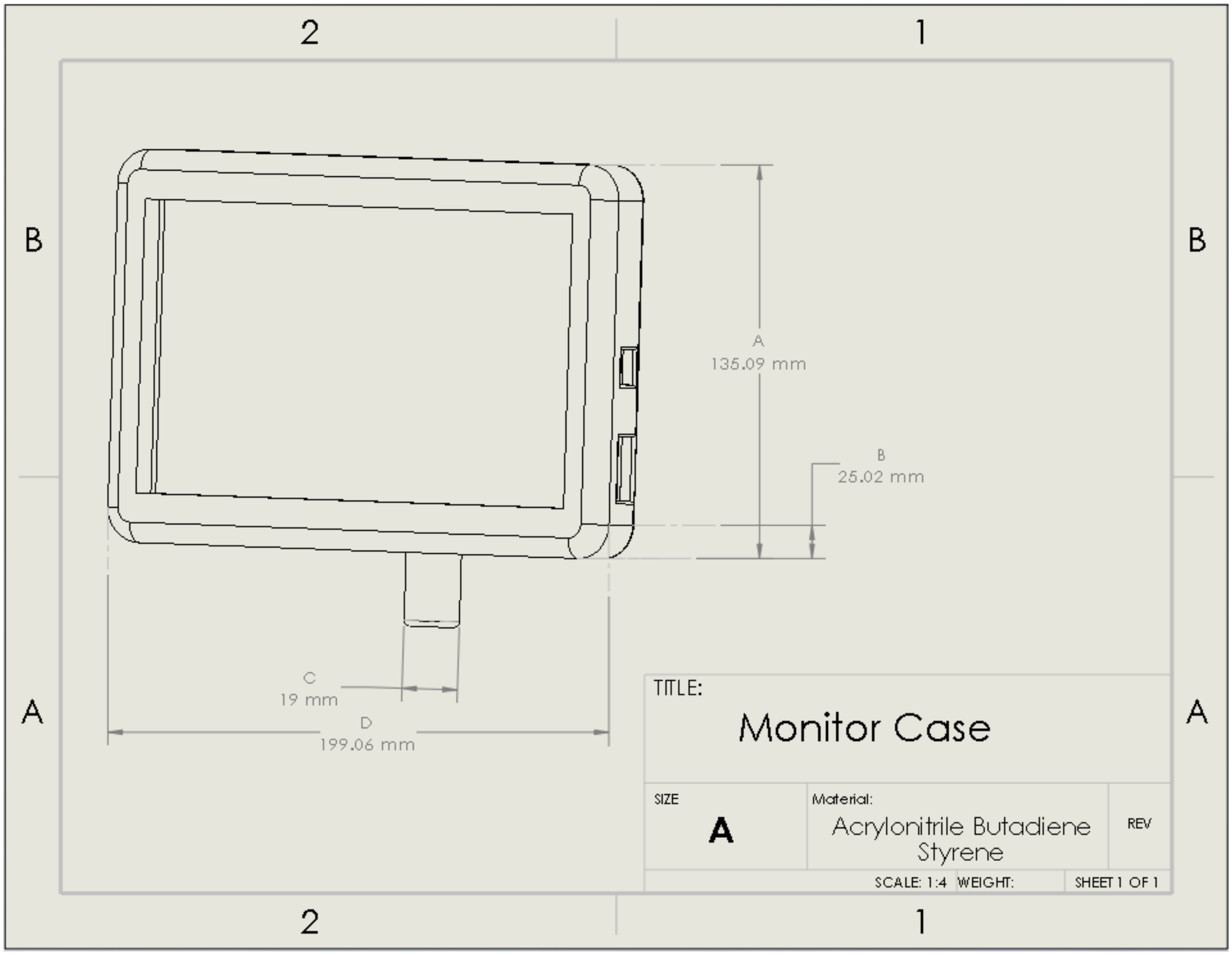


Figure 5.4.5. *A CAD drawing of the Monitor Encasement. (**a** and **d**) The height and width of the encasement. (**b**) The thickness of the encasement. (**c**)The required diameter of the holster to be position into a mic clip. The monitor encasement material is ABS. Monitor encasement weight is 306.22 grams.*

5.5. Test Data

Actual Distance(m)	Expected Visual Pathway Color (at Base of object)	Recorded Visual Pathway Color (at Base of object)
.5	Red	Red
1	Red	Red
1.5	Red (on-line)	Red (on-line)
2	Yellow	Yellow
2.5	Yellow	Yellow
3	Yellow	Yellow
3.5	Yellow (on-line)	Yellow (on-line)
4	Green	Green
4.5	Green	Green
5	Green (on-line)	Green

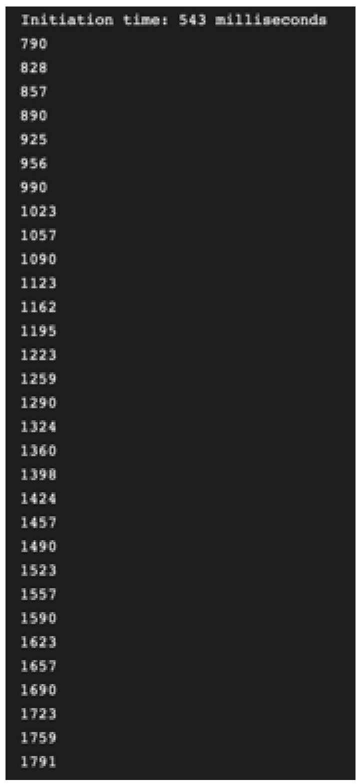
Table 5.5.1*: Results of visual pathway accuracy testing*

Trial Time [s.ms]	Output time on Monitor [s.ms]	Latency Time (i) [s]	Deviations (m-i)	Squared deviations (m-i)^2
10.67	10.36	.310	.001	10E-6
20.60	20.31	.290	.021	4.41E-4
30.38	30.05	.330	-.019	3.61E-4
40.13	39.84	.290	.021	4.41E-4
50.44	50.11	.330	-.019	3.61E-4
1:00.25	59.93	.320	-.009	8.1E-5
1:10.45	1:10.12	.330	-.019	3.61E-4
1:20.66	1:20.32	.340	-.029	8.41E-4
1:30.51	1:30.22	.290	.021	4.41E-4
1:40.90	1:40.60	.300	.011	1.21E-4
10.11	9.78	.330	-.019	3.61E-4
19.83	19.56	.270	.041	1.7E-3
30.05	29.75	.300	.011	1.21E-4
39.91	39.56	.350	-.039	1.5E-3
50.36	50.07	.290	.021	4.41E-4
1:00.11	59.78	.330	-.019	3.61E-4
1:10.16	1:09.81	.350	-.039	1.5E-3
1:20.47	1:20.16	.310	-.009	8.1E-5
1:30.14	1:29.87	.270	.041	1.7E-3
1:39.54	1:39.24	.300	.011	1.21E-4
n=20		**Avg Latency(m): .311**		**Sum of Squared Deviations: .0113**

***Table 5.5.2a**: Results from system latency testing with modified Visual Pathway*

Number of Trials	H_0	H_1	Average Latency	System Specification	Reject Null/Fail to Reject
20	P < 95%	P > 95%	311ms	300ms	Fail To Reject Null

Table 5.5.2b: *Statistical analysis (one Tailed-test) showing display monitor's Latency Confidence*



```
Initiation time: 543 milliseconds
790
828
857
890
925
956
990
1023
1057
1090
1123
1162
1195
1223
1259
1290
1324
1360
1398
1424
1457
1490
1523
1557
1590
1623
1657
1690
1723
1759
1791
```

Figure 5.5.1: *Sample Data of rs-imshow.cpp Output*

$\bar{y}$	$\hat{\sigma}$	t	*Result*
978 ms	*34.13ms*	*28.07*	*Reject Null*
19 fps	*0.62fps*	*-78.75*	*Reject Null*

Table 5.5.3: *Results for Initiation Time (Row 1) and Framerate (Row 2)*

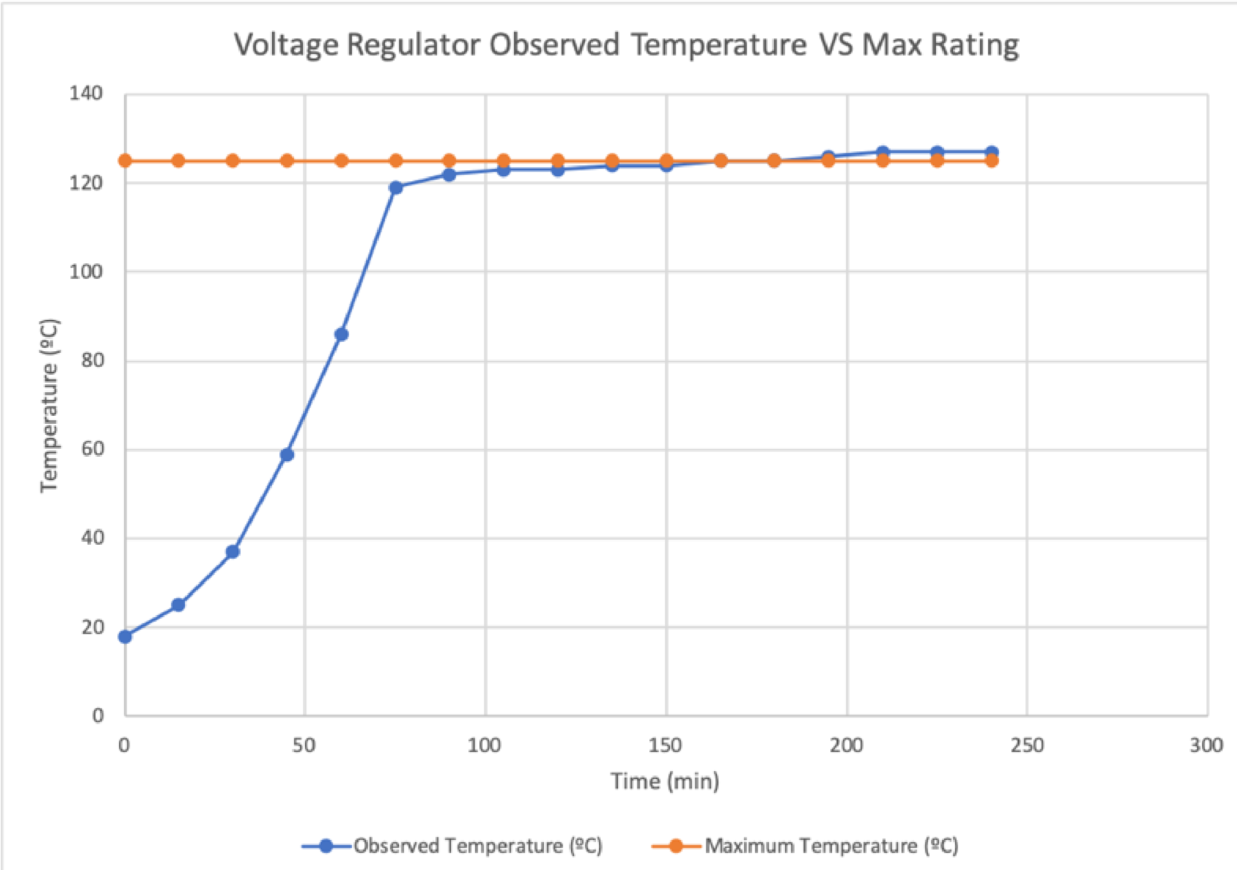


Figure 5.5.2: Heat Dissipation Data over 4 Hours VS Maximum Temperature Rating

	Simulation of ABS Material with ¼-20 Bolts	ABS Material Data Sheet	¼-20 Bolts
Tensile Strength	4.756e-002 MPa	42.5-44.8 MPa	N/A
Elongation at Break	2.495e-013	.23-.25	N/A
Factor of Safety (>=2)	N/A	N/A	1.1e+003

Table 5.5.4: *CAD encasement design material justification testing results.*

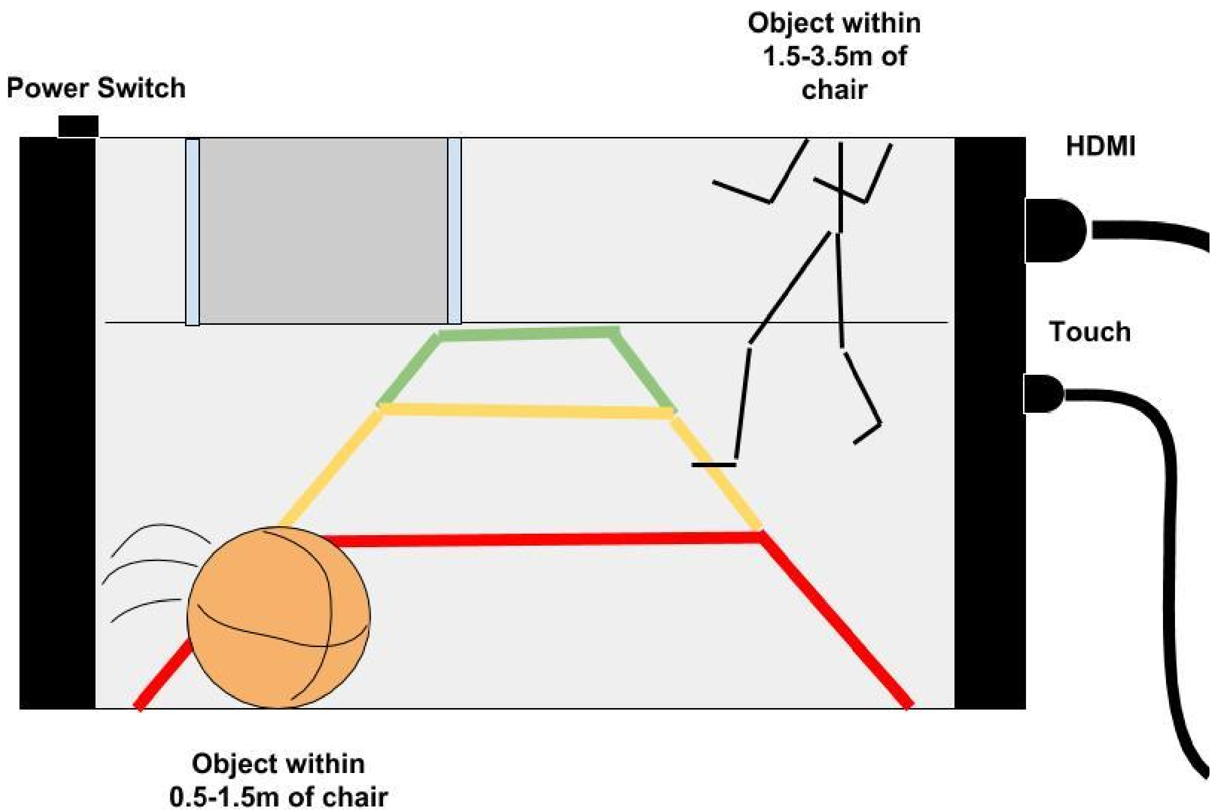


Figure.5.5.3 : *Vision for visual pathway overlay on the monitor display*

Number of Survey Participators	Can you see the display and its contents from this distance? (40 cm)	Can you see the display and its contents from this distance? (50 cm)
1	Yes	Yes
2	Yes	Yes
3	Yes	Yes
4	Yes	Yes
5	Yes	Yes
6	Yes	Yes
7	Yes	Yes
8	Yes	No (couldn't see in complete detail)
9	Yes	Yes
10	Yes	Yes
11	Yes	Yes
12	Yes	Yes
13	Yes	Yes
14	Yes	Yes
15	Yes	Yes
16	Yes	Yes
17	Yes	Yes
18	Yes	Yes
19	Yes	Yes
20	Yes	Yes
21	Yes	Yes
22	Yes	Yes
23	Yes	Yes
24	Yes	Yes
25	Yes	Yes
26	Yes	Yes
27	Yes	Yes
28	Yes	Yes
29	Yes	Yes
30	Yes	Yes

Table 5.5.5: Survey of 30 Participants for Display Monitor

Answer: Yes	Answer: No	Number of participants:	H_0	H_1	Z_{actual}	$Z_{critical}$	Reject Null/Fail to Reject
29	1	30	P < 95%	P > 95%	2.83	1.645	Reject Null

***Table 5.5.6**: Statistical analysis (Z-test) of Display Monitor*

H0	H1	N	Deg. Freedom	p-value	Crit. Value
μ = 499ms	μ > 499ms	4	3	0.05	2.353
μ = 30fps	μ < 30fps	20	19	0.05	-1.729

Table 5.5.7: *Statistical Hypothesis for Initiation Time (Row 1) and Framerate (Row 2)*

5.6. Technical Assembly Instructions.

In order to construct a replica of the back-up detection power wheelchair system. An Intel D435 Camera is required. Next 3D print a camera and monitor encasement based off of the solid works models in figures 5.4.2- to 5.4.5. Assure that the encasement can be directly attached to your power wheelchair or similar device. Measure the height at which your camera will be. To create a stream of optical data, refer to codes mentioned in the SDK and GitHub [10,11,14,15,21]. It is recommended that you use Rs-imshow and follow our guidelines for the creation of a visual pathway as shown in figure 5.3.2. To correctly align the visual pathway use trigonometry to calculate angle and distance from the camera based off of the height your camera is placed and cross-reference this with the size of your purchased monitor. After this correctly modify pixel color based off where you would like your visual pathway to lay. Assure that you then purchase an up board and required cables to connect it to both your monitor and camera. Before using your up board, image it with an operating system, we recommend Windows 10 or Linux. Your code for the visual pathway will initially be saved to a computer, once completed, if you are using C++, download your code as an executable file and upload it to your up board. Your up board also requires a battery for operation, assess how long you need your battery to function and how many volts and amp hours your up board will need, then purchase accordingly. Mount your camera to your encasement and your monitor to its encasement and then both encasements to your vehicle.

6. Bibliography

1) Bakhshi, Joe. “Wheelchair Facts, Numbers and Figures.” KD Healthcare Company. Jan. 2015.<https://kdsmartchair.com/blogs/news/18706123-wheelchair-facts-numbers-andfigures-i nfographic>
2) “Wheelchair Scooter Rear view Backup Camera System with a 3.5-Inch Screen.” Tadi Brothers. <https://www.tadibrothers.com/products/35-wheelchair-rearview-backup-camera-system>
3) “The C350 Corpus 3G: A class of its own.” Manual for C350 power chair. Permobil. 2015. <http://www.permobil.com/en/English/C/Products/C350-Corpus/>
4) Krishnamurthy, A. et al. “Powering automotive displays to create interactive driving experiences.” Texas Instruments. 2018. <http://www.ti.com/lit/wp/ssay003/ssay003.pdf>
5) “Medical electrical equipment -- Part 1-8: General requirements for basic safety and essential performance -- Collateral standard: General requirements, tests and guidance for alarm systems in medical electrical equipment and medical electrical systems” IEC 60601-1-8:2006 <https://www.iso.org/standard/41986.html>
6) “Batteries Carried By Airline Passengers: Frequently Asked Questions.” Federal Aviation Administration Office of Hazardous Materials Safety. Based on US DOT regulations (49 CFR, Sec. 175.10). Revised Sep. 9, 2016. PDF. <http://www.faa.gov/Go/PackSafe>
7) “Battery 101: The Difference Between a Trickle Charger and a Regular Battery Charger.” Northeast Battery. Web. <https://northeastbattery.com/trickle-charger-versus-regular-battery-charger/>
8) “Intel® RealSense™ Depth Camera D435” <https://click.intel.com/intelr-realsensetm-depth-camera-d435.html>
9) “Specifications.” UP: bridge the gap. UPDatasheetV8.5. <https://up-board.org/wp-content/uploads/datasheets/UPDatasheetV8.5.pdf>
10) “Rs-capture.cpp” Github. <https://github.com/IntelRealSense/librealsense/blob/master/examples/capture/rs-capture.cpp>
11) “rs-imshow.cpp” Github. <https://github.com/IntelRealSense/librealsense/blob/master/wrappers/opencv/imshow/rs-imshow.cpp>
12) “LM1084 5-A Low Dropout Positive Regulators.” Texas Instruments. Datasheet. PDF. <http://www.ti.com/lit/ds/symlink/lm1084.pdf>
13) “AWG Wire Gauges Current Ratings.” Engineering ToolBox, (2003). [online] Available at: <https://www.engineeringtoolbox.com/wire-gauges-d_419.html> [Accessed 05 Apr. 2019].
14) “librealsense” GitHub. <https://github.com/IntelRealSense/librealsense>
15) “opencv-3.4” GitHub. <https://github.com/opencv/opencv/tree/3.4>
16) *“*Information Technology Equipment – Safety – Part 1: General Requirements*”* UL Standard 60950-1, 2007.
17) “Requirements for Handling Electrostatic-Discharge-Sensitive (ESDS) Devices” JEDEC Standard JESD625B, 2011

18) “IEEE Standard on Video Techniques: Measurement of Resolution of Camera Systems” *1993 Techniques,* IEEE Standard 208, 1995. ISO 2580-1:2002
19) “PPI Calculator (Pixels Per Inch).” Omni Calculator. <https://www.omnicalculator.com/other/pixels-per-inch#how-to-calculate-ppi-the-theory.>
20) “Basic structures,” OpenCV. [Online]. Available: <https://docs.opencv.org/3.4/dc/d84/group__core__basic.html. [Accessed: 09-May-2019]>.
21) Rs-Distance <https://github.com/IntelRealSense/librealsense/blob/master/examples/C/distance/rs-distance.c>