# Development of a Telepresence System to Visually Assist in Navigating an Underwater Remotely Operated Vehicle

by

Umar B. Jara
Gahanna Lincoln High School
140 S Hamilton Road
Gahanna, Ohio 43230

ujara4645@gmail.com
(614) 432-2309

Prepared for Gahanna Lincoln High School
Science Academy Symposium
January 26, 2018

**The Development of a Telepresence System to Visually Assist In the Navigation of an Underwater Remotely Operated Vehicle**

**Umar Jara**
**Gahanna Lincoln High School, 140 S Hamilton Road, Gahanna, OH, 43230**

The purpose of this project was to develop a simple and cost-efficient telepresence prototype that could be used to assist in the visual navigation of underwater remotely operated vehicles [ROV]. ROVs were the application because of them being remotely controlled; immersive piloting would improve navigation.

This project operated by using accelerometer movements of a smartphone to control two servos. These servos were then affixed to a webcam so that the webcam could move up-and-down and left-and-right. Control of the servos via a smartphone were done through three components: an Arduino Uno microcontroller, a 1Sheeld shield, and a motor shield. The smartphone's accelerometer measurements were sent to the 1Sheeld via Bluetooth through its 1Sheeld Android application. These measurements were then sent to the Arduino and converted into servo rotation commands. The motor shield directly connected to both servos and executed the commands. In addition to smartphone control, a stereoscopic view of the webcam's image was achieved by using two sets of mirrored acrylic to split the webcam's image into a left and right image. These images were of the same subject but at two slightly different angles. The webcam's image streamed to a computer and then to the smartphone via a remote desktop application. The smartphone was then placed into a Google Cardboard Virtual Reality headset; the virtual reality headset recombined the two images into one, stereoscopic view.

The prototype proved promising. Angle conservation was measured between how much the phone was rotated up-and-down and how much the servos moved. Using a linear fit, the conservation had an $R^2 = 0.929$ for up and $R^2 = 0.927$ for down; this data showed a strong relationship between the phone and servo's rotations. Programming difficulties need to be overcome to consistently control the left-and-right servo. A cost analysis also should be performed to determine cost-effectiveness.

*Keywords:* Arduino, ROV, accelerometer, servo, telepresence

## Table of Contents

## Introduction

Underwater remotely operated vehicles [ROVs] are used for a variety of purposes. Some of these purposes include exploration, scientific inquiry, and education. ROVs can be used to explore shipwrecks, gather oil samples, and give a platform for Science, Technology, Engineering, and Mathematics [STEM] education to thrive. As insinuated by their name, ROVs are controlled remotely by a human operator usually on the surface or on a larger research vessel. The advantage of using ROVs over human divers or manned submersibles is that ROVs mitigate any risks humans may face. These risks may include extreme temperatures, tampering with underwater ecosystems, or various other environmental hazards. Essentially, ROVs allow humans to explore and understand the depths of the oceans from a safe vantage point.

Most ROVs, especially low-end and educational ones, are visually navigated by displaying one or more camera feeds to the human operator on one or more static screens. While this system usually suffices, many high-end and well-sponsored ROVs tend to use a system of telepresence. Telepresence allows for the operator to visually navigate the ROV by using virtual reality [VR] technology. VR works by taking two video streams of the same image but from slightly different angles. These images are then presented to the human eyes, individually, allowing the mind to combine them into a single image, thus creating a sense of depth in the image. Using VR technology, the operator is given an immersive experience which can possibly improve navigational performance. Thus, the reason most ROVs do not use telepresence is simply due to its cost and the complexity involved in integrating the system into an ROV.

The purpose of this study is to develop a cost-efficient and relatively simple telepresence prototype that can be used to assist in the visual navigation of underwater ROVs. This study intends to lower the bar of entry for low-end ROVs to use telepresence in their navigational systems.

# Review of Literature

## Virtual and Augmented Reality

VR is an experience that immerses its users into a virtual world. VR has application in a variety of fields, such as video games, simulations, and robotic control. VR works by displaying two 2-D [two-dimensional] images of the same subject to each eye, individually (Mullis, 2016). However, these images are from two slightly different angles. The brain then connects the two images together to create one 3-D [three-dimensional] image. This single image is also referred to as a stereoscopic display and is what creates the illusion of depth (Mullis, 2016). Augmented reality [AR] is similar to VR in the sense that both create immersive digital graphics. However, AR mixes these graphics with a vision of reality while VR creates an entirely virtual environment (Rouse, 2016).

The most primitive resemblance of modern VR/AR came in the year 1838 in the form of Sir Charles Wheatstone's "Stereoscope" (Wheatstone, 1838; "History Of Virtual Reality", n.d.). Wheatstone's research on presenting two 2-D pictures to the brain to create a 3-D image produced the principles that most, modern head-mounted displays [HMD] are based on today ("History Of Virtual Reality", n.d.). Innovation in the field of VR/AR began to take place in the 20th century due to advancements in electronics and computers. In 1929, Edward Link created the first commercial flight simulator known as the the "Link trainer" and sold several units to the United States military. Motors were used to control the device's pitch and roll and to simulate turbulence. William Gruber was among the first people to devise a home-product based on Wheatstone's "Stereoscope." Gruber's "View-Master" allows for stereoscopic viewing of select, still-images and was patented in 1939, still far preceding the advent of modern VR/AR HMDs.

The first VR/AR HMD with head tracking capabilities, known as “Headsight,” came in 1961 from engineers at the Philco Corporation (Axworthy, 2016; “History Of Virtual Reality”, n.d.). “Headsight” used magnetic head tracking and was connected to a closed-circuit television in order to remotely observe possibly dangerous situations. Up until 1968, all VR systems had displayed camera feeds or still-images. Ivan Sutherland and Bob Sproull changed the status quo via the advent of the “Sword of Damocles”. The “Sword of Damocles” was the first AR headset to display computer generated graphics to its users on top of a view of the real world (“The Sword of Damocles and the birth of virtual reality”, 2014; “History Of Virtual Reality”, n.d.). Although these graphics were quite simple and crude, this system is what defined the field of VR and AR (“The Sword of Damocles and the birth of virtual reality”, 2014; “History Of Virtual Reality”, n.d.). Throughout the 1990s, various attempts were made to popularize VR video games by arcades and video game companies, including Nintendo and SEGA (“History Of Virtual Reality”, n.d.). However, these movements never quite caught on and awaited a revival that would come a couple of decades later. Today, most consumer VR headsets are used for gaming and are capable of far more than their predecessors. VR/AR systems from corporations such as Facebook, HTC, Sony, Samsung, and Microsoft can give physical feedback, use gesture control, track movement with high accuracy, and so much more (Rigg, 2016; “History Of Virtual Reality”, n.d.).

**Google Cardboard**

Google Cardboard is Google’s way of bringing VR to the general public. Google Cardboard is less a specific HMD itself, but rather, a design that can be manufactured by practically anyone (Ripton, 2014). Cardboard works on the exact same principles as any other

VR/AR HMD, but its design is what makes Cardboard so special - not its technology. Use of components such as cardboard, magnets, velcro, rubber bands, lenses, and a smartphone is what makes Cardboard a starting point in VR for the average consumer. The simple design drastically lowers the cost of a VR experience when compared to various other market-standard HMDs; Cardboard headsets start out at five dollars while the Oculus base model starts at 400 dollars (Sinclair, 2017; Silbert 2017). While Cardboard does not compare in terms of performance to, say, an Oculus HMD, Cardboard fulfills its purpose of creating an inexpensive, entry-level headset. All graphics and head tracking software are housed within the smartphone connected to the Cardboard eliminating the need for extra hardware and external computers that most other HMDs require. Cardboard is compatible with a large variety of mobile applications including games built to be played in VR, video tours, movies, and even car test-drives (Ripton, 2014; Silbert, 2017). Google Cardboard opens up the usage and development of VR/AR technology to a whole new crowd of people by dramatically lowering the bar of entry to the field.

**Microcontrollers**

Microcontrollers are found in just about every product that requires some sort of electronic control from cars, to digital cameras, and even to microwaves (Brain, 2000). Microcontrollers act as a sort of central processing unit [CPU] for electrical circuits by receiving electrical signals, processing them, and emitting signals based on what is received (Dahl, 2013). Microcontrollers are similar to most computers in the sense that both can store variables, execute programs, and communicate with input and output [I/O] devices (Brain, 2000). While the two share some fundamental traits, slight nuances make each better for different purposes. In terms of memory, microcontrollers are often dedicated to a single purpose, so their programs are stored in read-only memory [ROM]. ROM consists of the device's programming that is essentially permanent; it stays on the microcontroller and is not written over even when it is turned off (Beal, n.d.). It is primarily for microcontrollers embedded inside of other devices where only a single program is ever executed. However, some microcontrollers - primarily those used for creating prototype electronics - can have new programs written on them as well as erase previous ones. While the types of inputs and outputs for microcontrollers and computers may vary, they work in the same way. Inputs are sent to the processor from an external device, and the processor then sends an output to another external device based on the original input. This process/concept can be seen in computers; for example, when a mouse is used to click open an application, the computer processes the input of the mouse being clicked while on the application and outputs a display of the application being opened. For microcontrollers, sensors are what give the microcontroller input data. An example of a sensor is an accelerometer - a device that measures dynamic and static acceleration - being used to turn a servo. A servo is a device that contains a motor shaft that can be programmed to have its shaft to be rotated to specific angular positions (Goodrich, 2013; "What is a Servo?", n.d.). A microcontroller, in this instance, would receive direction and magnitude of acceleration from the accelerometer. The microcontroller would then take these measurements and convert them to a direction and degree rotation. This new direction and degree measurement is sent to the servo as electrical signals which the servo executes. All these components of microcontrollers are customizable - allowing for easy specialization of a microcontroller based on the desired purpose of the electronic device.

**Arduino Uno**

Arduino is an open-source, prototyping utility made for users of all levels of electronics and programming ("What is an Arduino?", n.d.; "What is Arduino?", n.d.). The platform itself consists of two parts: hardware and software. Hardware comes in the form of the various microcontrollers that Arduino offers, the most common being the Arduino Uno. The Uno board packs a plethora of electronic components into a small space. One of these components is the power connection which can be the barrel jack that connects to a battery pack or a universal serial bus [USB] connection. Relating directly to the power supply of the board is its voltage regulator. A voltage regulator is fairly self-explanatory - it can be adjusted to a certain voltage cap and will limit the voltage that can flow through it. Another important component is the main integrated circuit [IC]. The main IC is where the processing of the microcontroller occurs. Most of the components on the board, with which interaction can occur, are different types of pins. These pins include voltage pins, ground [GND] pins, analog pins, digital pins, pulse-width modulation pins, and analog reference pins [AREF]. Voltage pins come as five volts and 3.3 volts and are used to supply that amount of electricity respectively. GND pins acts as a ground and are integral to the completion of every circuit as the GND completes the circular flow of electricity that starts from the voltage source. Analog pins are used to read input data from sensors, such as temperature or light sensors, ass opposed to digital pins, which directly read electrical signals such as a button being pushed. Digital pins can also be used for controlling digital outputs such as playing sound on a piezoelectric buzzer or illuminating a light-emitting diode [LED]. PWM is a type of digital signal that allows for the control of the amount of time that the signal has full voltage and when it does not have any voltage ("Pulse Width

Modulation", n.d.). Power received by the signal is proportionate to the amount of time that the signal has full voltage while the voltage is being turned on and off. Thus, if the voltage entering is five volts, and PWM turns the voltage on for 50% of the time, then the voltage being emitted is 2.5 volts ("Pulse Width Modulation", n.d.). No pins exist on the Uno that are exclusively for PWM, but some digital pins can be used for this purpose. The last type of pin that can be found is the AREF. AREFs can be used to limit the voltage of analog input pins but are actually rarely used.

An equally important part of the Arduino platform is the software. The software is used to program the board to execute the desired functions. Available on Mac, Windows, and Linux, an integrated development environment [IDE] is the application in which the program is written (Heath, 2017). The language that is written inside the IDE is a simplified variation of C and C++ programming languages (Heath, 2017; "Arduino Software", n.d.). C was created in the 1970's, and its original purpose was to write Unix - a computer operating system - applications but now is used to write programs for many different operating systems ("C/C++", 2007). Some advantages of C, relative to the languages preceding it, is that it is memory-efficient, easy to understand, and capable of more robust software programs.. C++ is based on C and uses similar syntax but is different in that C++ is object-oriented. Object-oriented programming is when the functions and variables inside of a program can be defined so that they can be referred to later ("C/C++", 2007; Beal, n.d.). Switching to object-oriented programming creates a substantial efficiency boost and makes creating more complex programs significantly easier. Possibilities are endless when it comes to Arduino due to the combination of the numerous different electronics components available and the robustness of the programming language.

**Underwater ROVs**

Underwater ROVs have an assortment of different purposes; ROVs can be used for deep-sea exploration, collecting data for scientific experiments, and even promoting STEM education. An ROV is a robot that is controlled remotely - usually by a human on land or on a research ship nearby. ROVs give the distinct advantage of removing human divers from danger (Rouse, 2011). For example, if measurements or data needed to be collected several hundred feet deep in the ocean, then an ROV can be deployed instead of a human, so the diver can avoid having to endure the high pressures of such a great depth. Another scenario where ROVs may be useful include exploring shipwrecks. For example, the Titanic shipwreck was discovered in 1985 and was explored by an ROV named Jason Junior in 1986 (“Robots reveal Titanic secrets”, 2012). Many ROVs are outfitted with sensors capable of measuring variables such as temperature and water clarity (“What is an ROV?”, n.d.). Information gained about the Earth’s oceans allows scientists to better understand how the environment is changing and what effect humans may be having on it. In addition to passive use, underwater robots can be used for active operations by using manipulators, claws, or other tool packages. In 2010, during the Deepwater Horizon oil spill, multiple ROVs were used to stop oil leakage (Rioux, 2010). The robots streamed live, high-definition video of them working non-stop, performing maintenance on BP’s wells to stop oil from gushing out. Use of the ROVs was vital to damage control and preventing the oil well from further destroying the environment of the Gulf of Mexico. Usage of underwater ROVs has clear benefits for both humans and the environment.

**Telepresence**

Telepresence involves the use of VR technology or methods to control robots (Rouse, 2015). A robot is typically controlled over large distances by a human user, and the robot's view is also displayed to the human user. While some ROVs capitalize on using VR technology to create an immersive piloting experience, many resort to using images streamed from the ROV's camera(s) to the user's monitor(s) (Rouse, 2011). The rationale of using telepresence is similar to that of using ROVs; telepresence removes humans from any imminent danger of the situation or can even possibly improve the performance of a specific task (Rouse, 2015). Bomb-defusal bots are an example of removing humans from a dangerous, but necessary situation. Military or law enforcement personnel operating the robot are removed from the danger of the explosion but are still able to defuse the explosive by using the robot's manipulators (Rouse, 2015). A method of improving human performance is through the use of surgical robots. Robotic surgical tools have high degrees of flexibility and control and are relatively small - allowing for minimally invasive surgeries ("Robotic surgery", n.d.). Using VR technology to assist in the operation of robots could have an overall positive effect on robotic performance.

# Methods and Materials

Creating a telepresence navigational system for an ROV consisted largely of four parts. The first part was assembling the electronic components that connected the smartphone to the Arduino and the Arduino to the servos. The second part was to program the software to tell the Arduino how to interact with the servos based on the input it receives from the smartphone. Parts three and four involved 3D printing the image splitting apparatus and a construction to affix the camera and servos together.

## Materials

Materials were chosen based on their cost effectiveness and ease of accessibility. Therefore, most components were purchased online or reused from previous projects. For example, the Arduino Uno used was an Elegoo UNO clone salvaged from a hobby electronics starter kit **[Figure 1]**. Connected to the Arduino were two shields: the 1Sheeld **[Figure 2]** and a motor shield **[Figure 3]** . The 1Sheeld provided a way to wirelessly transmit sensor data from a smartphone to an Arduino Uno, and the motor shield allowed for easy connection to the servos from the Arduino Uno. Two Futaba S3003 180° servos used were the servos used to rotate the webcam **[Figure 4]**. Both the image splitting apparatus and camera-servo attachment were 3-D printed using Polylactic Acid [PLA] plastic filament. All 3D printing was done on a MakerGear M2 printer. Mirrored acrylic operated as the mirrors inside of the apparatus. The webcam attached to the apparatus was a c615 Logitech HD Webcam **[Figure 5]**. An ASUS UX305C was the computer used to relay the video from the webcam to smartphone. The smartphone from which the accelerometer data is collected and video is displayed was a Google Pixel; the data was transmitted from the phone to the 1Sheeld via the 1Sheeld Android mobile application. An

EYEKOP Virtual Reality Viewer was the Google Cardboard clone used to recombine the split image from the webcam into a stereoscopic view **[Figure 6]**.

**Assembling Electronics**

The prototype focused around the combination of shields used and their interactions. An Arduino Uno, 1Sheeld, and motor shield were connected by placing the Uno at the bottom, connecting the 1Sheeld to its pins with the pins underneath the 1Sheeld, and the motor shield on top of the 1Sheeld in the same fashion **[Figure 7]**. The two servos were connected via their ribbon wires into the two designated servo pins on the motor shield. The webcam plugged into the computer via USB and streamed its image to the computer through a Logitech webcam software. Connecting the smartphone to the 1Sheeld was done via the 1Sheeld Android app where it connected to the shield with Bluetooth and also transmitted its data through Bluetooth.

**Programming Software**

Software was designed in the Arduino IDE and was used to input data that described the smartphone's orientation and acceleration and to output servo movements **[Figure 8]**. The 1Sheeld app reads the smartphone's sensor data and transmits it via Bluetooth to the 1Sheeld. From the 1Sheeld, the Uno takes the acceleration that the smartphone experiences from the z and y axes and maps it as a value from zero to 180. By mapping, the Arduino takes the inputted range of values and fits it to a desired range of values. In this scenario, one would want to take the range values that the phone's accelerometer reads and fit them onto a range from zero to 180,so that an accelerometer reading can be turned into a degree of servo movement. The z-axis movements correlated to the servo that rotate the camera up-and-down and the y-axis to the servo that rotate the camera left-and-right. Y-axis angles had to be multiplied by ten because the phone being in a landscape orientation produced y-axis readings that were between approximately three and negative three - including decimals. This number is a problem because the range of accelerometer movements is relatively small due to the mapping function truncating decimals. What this means is that if the accelerometer readings come out to be in a range between, say, negative two and two, then the Arduino will only give five possible positions for the servo to rotate to because the integers of the range of accelerometer readings would be negative two, negative one, zero, positive one, and positive two. The newly calculated angle is the position the servos moved to.

**Image Splitter and Camera Attachment**

The image splitting apparatus was used to take an image and split it into two. These two images were of the same subject but at two slightly different angles; one image represents the view the right eye has and the other represents the left; VR viewers require this kind of split-image in order to create a stereoscopic view. The prototype was modeled in Autodesk Inventor and 3-D printed with PLA plastic filament. It was designed around the dimensions of the webcam so that it could be easily affixed and had an appropriately-sized viewing window. Multiple iterations were printed in order to create a prototype well-suited for the camera's viewing angle and placement for the pieces of mirrored acrylic **[Figure 9]**. Mirrored acrylic is used so that the image from the viewing window is split into two images. See **Figure 10** for a diagram of how the pieces of mirrored acrylic were placed. The camera attachment was also 3-D printed from PLA plastic filament and had to go through multiple iterations in order to produce a version that more appropriately fitted the servo **[Figure 11]**.

**Testing and Data**

Testing and data was collected mainly for two figures: angle conservation and cost. Angle conservation was tested to determine if moving the phone a certain amount of degrees made the servo rotate that same amount of degrees. An ideal prototype would have a ratio of one-to-one in terms of phone rotation on a certain axis to the servo's rotation corresponding to that axis. This ratio was found by making a servo stationary and affixing two pens onto the servo horn in a fashion that allowed for one of the pens to mark on a piece of paper placed underneath the servo. Then, the phone was rotated ten degrees at a time on the axis that corresponded to the stationary servo. Every ten degrees that the phone moved, a mark was made on the piece of paper where the pen ended its movement **[Figure 12]**. In the end, the angle was measured at each mark from the starting point and compared to the increment of ten degrees that the servo should have ideally rotated.

In addition to testing technical ability, the prototype's economic effect was considered. The cost of the prototype was found by taking the sum of all costs. For the mirrored acrylic, the cost was found by taking the ratio between the size of the full sheet and the sum of the size of all the pieces and multiplying them by the cost of the full sheet. 3D printing costs were found by multiplying the sum of the masses of the final iterations of the image splitter and camera-servo attachment by the mass-to-dollar ratio of the PLA filament. All other components of the prototype were ordered per unit.

## Discussion

Angle conservation tests were planned to be done on both the up-and-down and left-and-right servos. Due to technical difficulties, however, only the up-and-down servo was operating in a condition deemed to be satisfactory for testing. **Figure 13** contains the graph that plots the phone's rotation up on the x-axis and the servo's rotation on the y-axis. **Figure 14** shows the same test on the same servo but in a downwards rotation. **Table 1** displays exact values of both of the servo's rotations during the angle conservations tests. The equation of the servo's upwards rotation came out to be $y = 0.88x + 8.4$ with an $R^2 = 0.929$; the downwards rotation's equation was $y = 1.07x + 15.7$ with an $R^2 = 0.927$. **Table 2** contains a cost analysis of the prototype; the table includes each item, its quantity, price, and total costs. The total cost of the prototype was $126.43.

# Conclusion

Development of a system using telepresence to visually assist in the navigation of an ROV focused around cost-efficiency and angle input to angle output accuracy.

## Angle Conservation

Accuracy was tested for in the angle conservation tests. Ideally, the slopes of the lines that represented the relationship between how much the phone rotated on a particular axis and the rotation of its corresponding servo would be one. This number is ideal because a one-to-one ratio means that the servo rotates exactly one degree for every one degree the phone rotates. Unfortunately, this ratio was not quite achieved. For upwards rotation, the slope was 0.88; for downwards rotation, the slope was was 1.07. While neither are quite one, both are rather close - especially the downwards rotation. Another point of interest on the graphs is the extremes. The upwards rotation graph seemed to flatten out between 70° and 90° which indicated that the servo ceased to make noticeable movement, and its range of movement upwards has been exhausted. This trend may be affected by where the starting point for the servo was located, or it may just be a manufacturing error in the servos. The downwards rotation graph displayed a similar issue where an area of the graph stopped changing in y-value.  For the downwards rotation graph, however, a cease in movement occurred between 0° and 20°. The cause of this abnormality is slightly more ambiguous and may also just be due to a manufacturing error in the servos. Another major component of the data lies within the angle conservation testing being done on the right-and-left servo. However, due to the nature of the 1Sheeld application, it was not programmed to be in a landscape orientation. This format created problems with moving the phone left-and-right because the only axis that seemed to be affected by left-and-right movement also seemed to be affected by acceleration on other axes. Multiple iterations of the program were made that varied in usage of the phone's orientation sensor, gyroscope, and accelerometer, but no clear solution was found while using one of the aforementioned sensors or combinations of them. Therefore, reliable and consistent data could not be presently collected for left-and-right rotations.

**Cost Efficiency**

Creating a cost-efficient prototype was achieved by the usage of inexpensive and easily-available materials. In the cost analysis, it is seen that the majority of the cost came from the 1Sheeld Arduino shield. With the shield, the total cost of the prototype totaled $126.43; excluding the 1Sheeld, the cost totals $76.43; the 1Sheeld comprised of approximately 40% of the total cost. The smartphone was not considered in the costs because of the fact that any Google Cardboard-compatible device would suffice; it is also assumed that the smartphone is already owned as the usage of smartphones has become ubiquitous. If a smartphone is not owned, an additional $100 to $1000 can be added to the total cost. At the system's current costs, its cost is over three times less than that of the Oculus Rift's; a VR headset and interactive system used by the likes of the Massachusetts Institute of Technology in their robotic remote operation research.

In the future, further work needs to be done on methods of consistently and reliably moving the left-and-right servo so that testing can be performed on the servo's angle conservation. More trials also need to be done in order to determine if there is any statistical significance between the slopes of upwards and downwards rotations. Costs should also be cut by substituting components. The Futaba S3003 servos can be substituted with servos of the same function but of a different manufacturer. The same substitution can be performed with the Logitech C615 webcam; all that is required of the camera is one that is capable of streaming its picture to a computer via USB. It is also possible to bring down 3D printing costs by exploring other types of plastic filaments. A part of finalizing the system would be to entirely waterproof it. Waterproofing can be done by sealing the servos in plastic dip and installing o-rings; the camera can be waterproofed by stripping it down to its bare components and potting these components inside of epoxy. Servos and the camera are the main points of concern as they would be the only two components under the water. While much work still needs to be done to finalize the prototype, it has shown much promise in its fundamental principles of operation.

## Ethical Considerations

Development of a telepresence system to assist in navigation of an ROV brings many benefits but also has concerns. One of the greatest benefits of the prototype is educational. Being able to assemble a system without extremely large expenditures - compared to other commercially available VR platforms - opens new doors for groups without large funds. Schools would be able to utilize the system to foster more immersive educational experiences such as when using underwater robots for the Marine Advanced Technology Education [MATE] competition. Another educational experience could be taking an ROV to local bodies of water and observing in order to have a better grasp on underwater ecology.

However, this technology can be exploited for malicious purposes. One such purpose is naval pirating; individuals may use the system to look for and attack unsuspecting civilians at sea. The system can give these individuals an advantage in their scouting methods, or if an ROV has manipulators, it can increase opportunities for destruction of property. In the end, it is the responsibility of the user to responsibly use this prototype.

**Acknowledgements**

Without the support of a few particular people, this project would not have been nearly what it is now. A teacher of mine, Mr. Kunselman, shared with me a great amount of 3D printing knowledge and allowed me use of his 3D printing machines. It was with his support that I was capable of quickly prototyping different iterations and fabricate major components of my project. I would like to thank Mr. Donelson, a former teacher here at GLHS, for encouraging me to pursue this project idea when I was originally bouncing ideas off him. Finally, I would like to thank my instructors, Mr. Bruns and Mrs. Rice, for all that they have taught not only me but our entire class. Mr. Bruns and Mrs. Rice have given students skills and knowledge that they traditionally would not have gained until college. This knowledge has given us an edge up on our peers and have allowed us to begin flourishing in whichever field we choose to pursue. Their course challenges students by forcing them to be better scientists and engineers.

# References

Accelerometer Shield acts as Arduino Shield via 1Sheeld. (n.d.). Retrieved from https://1sheeld.com/shields/accelerometer-shield/

Arduino Software. (n.d.). Retrieved from https://www.arduino.cc/en/Main/FAQ#toc11

Axworthy, J. (2016, April 01). The origins of virtual reality. Retrieved from https://www.wareable.com/wearable-tech/origins-of-virtual-reality-2535

Beal, V. (n.d.). OOP – Object Oriented Programming. Retrieved from http://www.webopedia.com/TERM/O/object_oriented_programming_OOP.html

Beal, V. (n.d.). ROM - read-only memory. Retrieved from http://www.webopedia.com/TERM/R/ROM.html

Bitar, H. (n.d.). Bluetooth-controlled Pan/Tilt Servo Platform Using Android As Remote. Retrieved from http://www.techbitar.com/bluetooth-controlled-pan-tilt-servo.html

Boddington, D. (2017, January 23). Virtual reality: Recognising the risks. Retrieved from http://www.sciencefocus.com/article/future/virtual-reality-recognising-risks

Brain, M. (2000, April 01). How Microcontrollers Work. Retrieved from http://electronics.howstuffworks.com/microcontroller.htm

C/C++. (2007, October 11). Retrieved from https://techterms.com/definition/cplusplus

Charara, S. (2017, May 22). Explained: How does VR actually work? Retrieved from https://www.wareable.com/vr/how-does-vr-work-explained

Dahl, Ø N. (2016, December 13). What is a Microcontroller? Retrieved from https://www.build-electronic-circuits.com/what-is-a-microcontroller/

Goodrich, R. (2013, October 01). Accelerometers: What They Are & How They Work. Retrieved from https://www.livescience.com/40102-accelerometers.html

Heath, N. (2017, March 27). Arduino: The smart person's guide. Retrieved from http://www.techrepublic.com/article/arduino-the-smart-persons-guide/

History Of Virtual Reality. (n.d.). Retrieved from https://www.vrs.org.uk/virtual-reality/history.html

Madden, B. (2011, October 09). Google launches "Chrome Remote Desktop," full desktop sharing via Chrome. What does this mean? Retrieved from http://www.brianmadden.com/opinion/Google-launches-Chrome-Remote-Desktop-full-desktop-sharing-via-Chrome-What-does-this-mean

Mullis, A. (2016, July 15). How does virtual reality work? Retrieved from http://www.androidauthority.com/virtual-reality-work-702049/

NOAA Ocean Explorer: Technology: Communication Technologies: Telepresence Technology. (2013, April 11). Retrieved from http://oceanexplorer.noaa.gov/technology/commstech/telepresence/telepresence.html

Pan Tilt Controlled by Cell Phone. (2016, November 3). Retrieved from https://create.arduino.cc/projecthub/lagsilva/pan-tilt-controlled-by-cell-phone-e8c3a3

Perez, S. (2014, April 16). Google Launches Chrome Remote Desktop On Android, Allowing Mobile Access To Your PC. Retrieved from https://techcrunch.com/2014/04/16/google-launches-chrome-remote-desktop-on-android-allowing-mobile-access-to-your-pc/

Pulse Width Modulation. (n.d.). Retrieved from https://learn.sparkfun.com/tutorials/pulse-width-modulation

Ray, M. (2016, May 13). MuonRay. Retrieved from http://muonray.blogspot.ie/2016/05/remote-augmented-reality-and-virtual.html

Rigg, J. (2016, June 01). Teslasuit does full-body haptic feedback for VR. Retrieved October 1, 2017, from https://www.engadget.com/2016/01/06/teslasuit-haptic-vr/

Rioux, P. (2010, July 16). Undersea robots are heroes of Gulf of Mexico oil spill fight. Retrieved from http://www.nola.com/news/gulf-oil-spill/index.ssf/2010/07/undersea_robots_are_heroes_of.html

Ripton, J. (2014, December 18). Google Cardboard: Everything you need to know. Retrieved from http://www.techradar.com/news/phone-and-communications/mobile-phones/google-cardboard-everything-you-need-to-know-1277738

Ripton, J. (2014, December 18). Google Cardboard: Everything you need to know. Retrieved October 2, 2017, from http://www.techradar.com/news/phone-and-communications/mobile-phones/google-cardboard-everything-you-need-to-know-1277738

Robotic surgery. (2017, June 06). Retrieved from http://www.mayoclinic.org/tests-procedures/robotic-surgery/basics/definition/prc-20013988

Robots reveal Titanic secrets. (2012, April 17). Retrieved from http://www.euronews.com/2012/04/17/robots-reveal-titanic-secrets

Rouse, M. (2011, August). What is underwater ROV (underwater remotely operated vehicle)? - Definition from WhatIs.com. Retrieved from http://whatis.techtarget.com/definition/underwater-ROV-underwater-remotely-operated-vehicle

Rouse, M. (2015, August). What is telepresence? - Definition from WhatIs.com. Retrieved from http://whatis.techtarget.com/definition/telepresence

Rouse, M. (2016, February). Augmented Reality (AR). Retrieved October 1, 2017, from http://whatis.techtarget.com/definition/augmented-reality-AR

Sensors Shields. (n.d.). Retrieved from https://1sheeld.com/tutorials/sensors-shields/

Silbert, S. (2017, August 22). All About Cardboard, Google's Low-Cost Virtual-Reality Device. Retrieved from https://www.lifewire.com/cardboard-googles-low-cut-virtual-reality-device-3441416

Sinclair, P. (2017, July 14). Oculus Rift vs. Google Cardboard, Which is the Better VR Experience? Retrieved from https://www.allhomerobotics.com/oculus-rift-vs-google-cardboard/

The Sword of Damocles and the birth of virtual reality. (2014, March 19). Retrieved from http://simpublica.com/2014/03/19/the-sword-of-damocles-and-the-birth-of-virtual-reality/

The Tools of Ocean Exploration. (n.d.). Retrieved from http://www.nautiluslive.org/tech

What is a Servo? (n.d.). Retrieved from https://www.servocity.com/what-is-a-servo

What is an Arduino? (n.d.). Retrieved from https://learn.sparkfun.com/tutorials/what-is-an-arduino

What is an ROV? (n.d.). Retrieved from http://oceanexplorer.noaa.gov/facts/rov.html

What Is Arduino? (n.d.). Retrieved from https://learn.sparkfun.com/tutorials/what-is-an-arduino

What is Arduino? (n.d.). Retrieved from https://www.arduino.cc/en/Guide/Introduction

Wheatstone, C. (1838). Contributions to the Physiology of Vision.-Part the First. On some remarkable, and hitherto unobserved, Phenomena of Binocular Vision. *The Scientific Papers of Sir Charles Wheatstone,* 225-283. doi:10.1017/cbo9781139057950.026

## Appendix


**Figure 1:** Arduino Uno Eleego clone


**Figure 2:** 1Sheeld Arduino shield

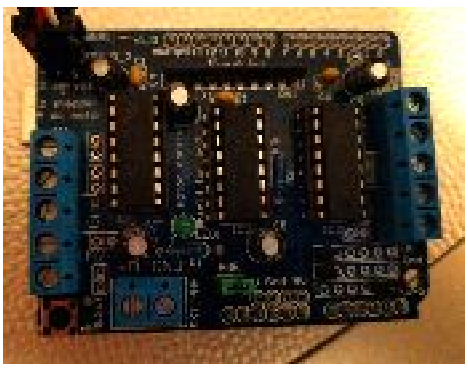

**Figure 3:** Arduino motor shield

**Figure 4:** Futaba S3003 180° servos

**Figure 5:** Logitech c615 webcam

**Figure 6:** Front and back view of the EYEKOP Virtual Reality Viewer

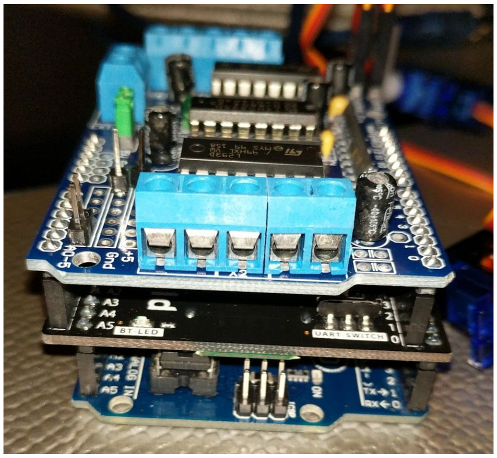

**Figure 7:**(Up to down) Motor shield, 1Sheeld, and Arduino Uno

```
#define CUSTOM_SETTINGS
#define INCLUDE_ACCELEROMETER_SENSOR_SHIELD

#include <OneSheeld.h>     //Include 1Sheeld Library
#include <Servo.h>         //Include Servo Library

Servo myservoY;            //Create servo object to control Orientation Y
Servo myservoZ;            //Create servo object to control Orientation Z

double angleY, angleZ;     //Declraing calibration values
double angleZi = 0;        //Used to calibrate angles values of each servo
double angleYi = 0;
boolean k = true;

void setup()
{

  OneSheeld.begin();
  AccelerometerSensor.setOnValueChange(readAngleXi);

  myservoY.attach(9);    //Servo2 in the Motor Shield - connected to Analog Pin 09 of Arduino
  myservoZ.attach(10);   //Servo1 in the Motor Shield - connected to Analog Pin 10 of Arduino
  myservoY.write(90);    //ServoY test
  delay(1000);
  myservoY.write(0);
}
void loop()
{
  angleY = AccelerometerSensor.getY();       //Ready Y axis accelerometer data
  angleY = (angleY + angleYi) * 10;          //Multiply angle value by 10 because the agle value is too small so when mapping,
  angleY = map(angleY,-15, 15, 0, 180);      //decimals don't get truncated or else there will only be ~6 positions the servo can move to
  myservoY.write(angleY);                    //Write new angle to servo
  delay(15);

  angleZ = AccelerometerSensor.getZ();
  angleZ = map(angleZ, -10, 10, 0, 180);     //Maps accelerometer's range of value to the servo
  angleZ = angleZ + angleZi;
  myservoZ.write(angleZ);
  delay(15);
}

void readAngleXi(float YAxis, float ZAxis, float XAxis)
{
  if (k) {
    angleYi = -YAxis;
    k = false;
  }

}
```

**Figure 8:** Program in Arduino IDE with comments

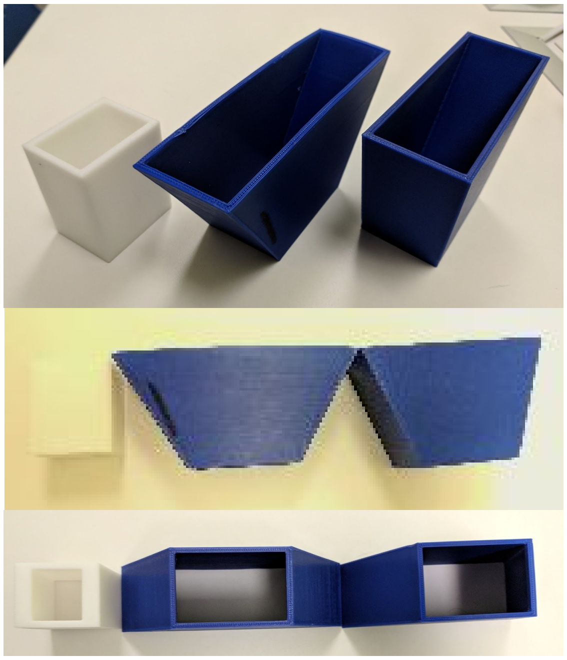

**Figure 9:** Angled top, bottom, and side view of three different iterations of the image splitter

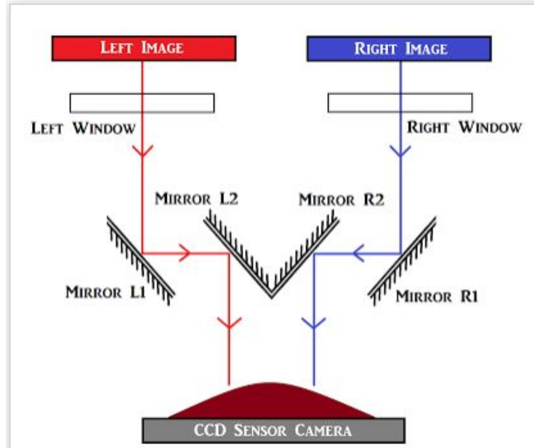


**Figure 10:** A diagram of how the mirrors are arranged in the image splitter (Ray, 2016)

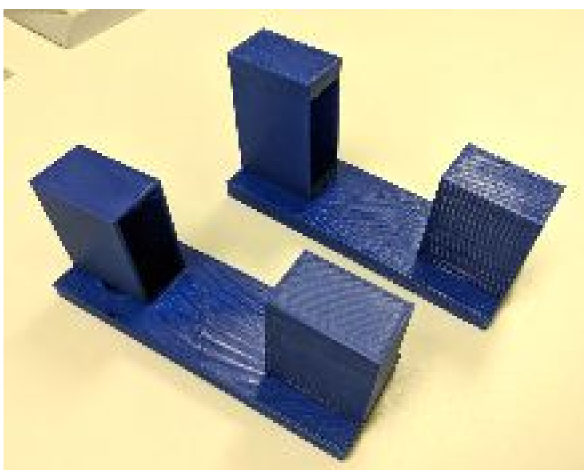

**Figure 11:** Angled top view of two iterations of the servo-camera attachment

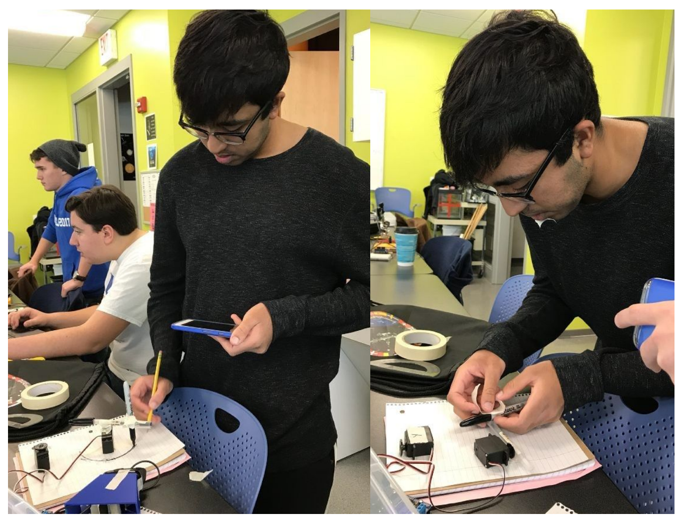

**Figure 12:** Setting up for measuring angle conservation

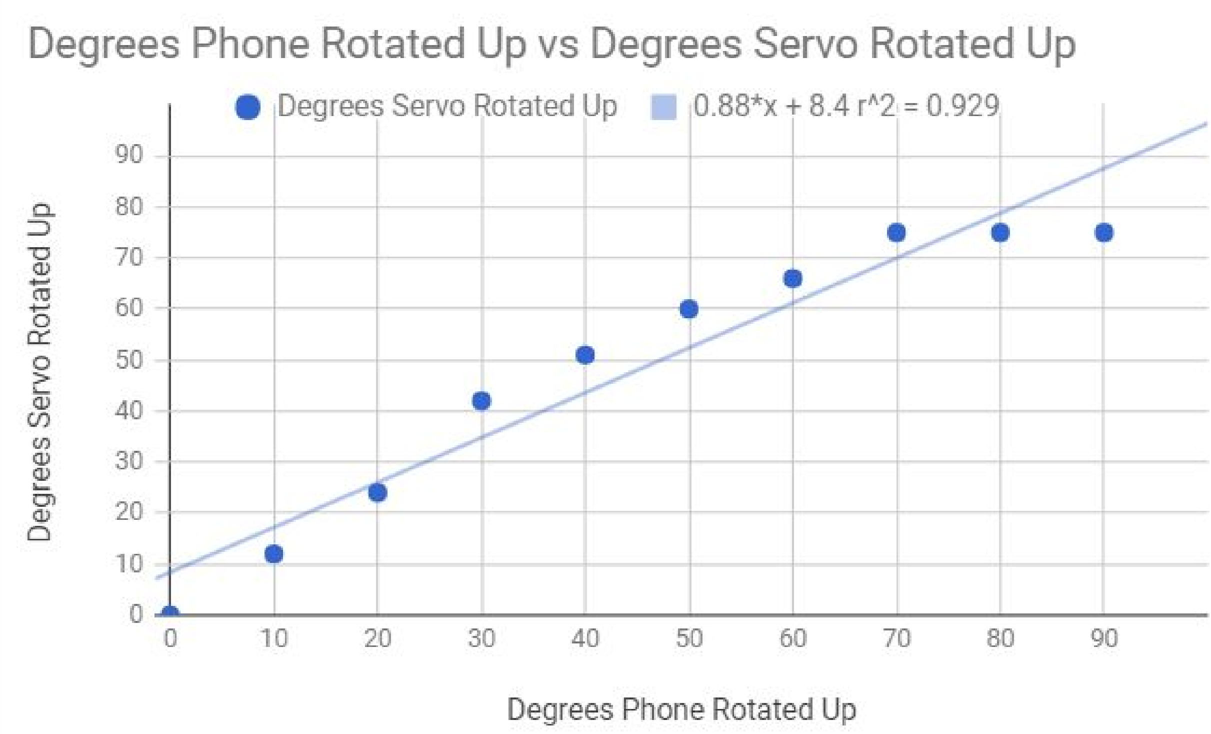


**Figure 13:** Angle conservation test on upwards rotation

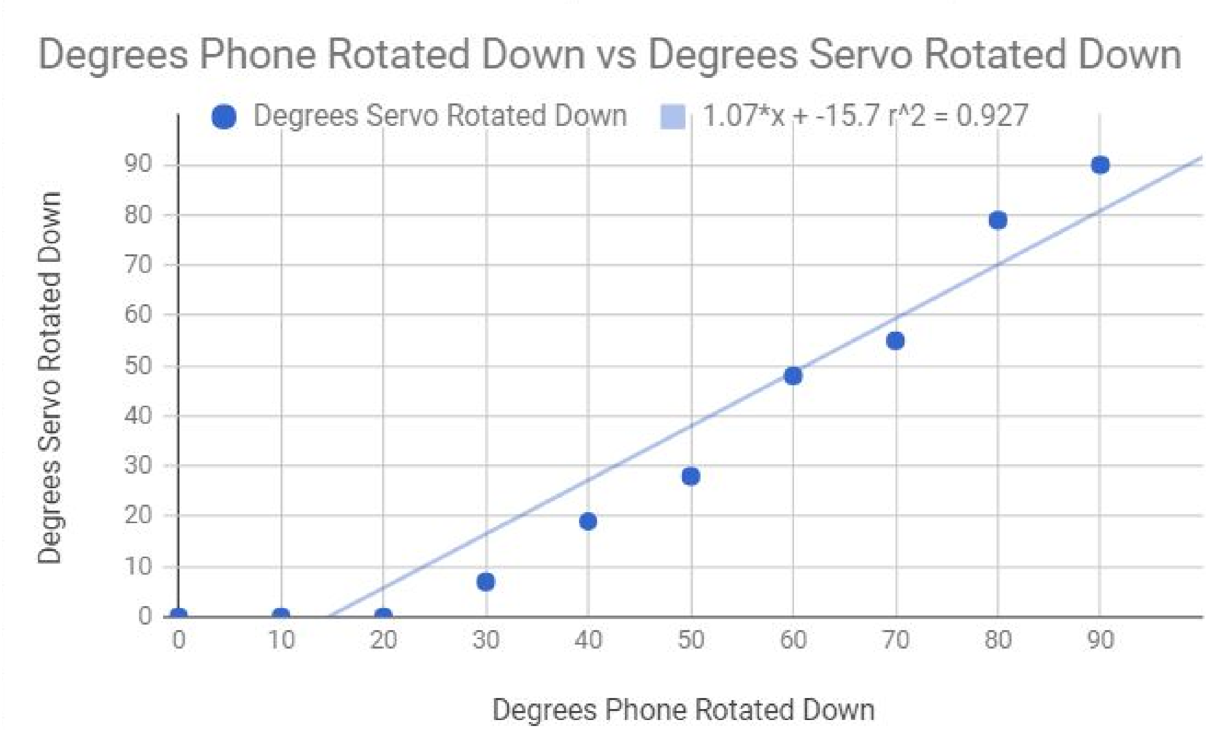


**Figure 14:** Angle conservation test on downwards rotation

| Degrees Phone Rotated From Starting Point | Degrees Servo Rotated Up | Degrees Servo Rotated Down |
|---|---|---|
| 0 | 0 | 0 |
| 10 | 12 | 0 |
| 20 | 24 | 0 |
| 30 | 42 | 7 |
| 40 | 51 | 19 |
| 50 | 60 | 28 |
| 60 | 66 | 48 |
| 70 | 75 | 55 |
| 80 | 75 | 79 |
| 90 | 75 | 90 |

**Table 1:** Values of the angle conservation on the up/down servo

| Item | Quantity | Price | Cost |
|---|---|---|---|
| Arduino Uno | 1 | $11.00 | $11.00 |
| Google Cardbord | 1 | $6.00 | $6.00 |
| Futaba S3003 servo | 2 | $12.74 | $25.48 |
| Logitech C615 webcam | 1 | $32.00 | $32.00 |
| PLA | 79.8 grams | $0.023 / gram | $1.84 |
| 1Sheeld Arduino Shield | 1 | $50.00 | $50.00 |
| Mirrored Acrylic | 4 1.375" x 1" | $10.75 / sq. ft. | $0.11 |
| Total Cost: | | | $126.43 |

**Table 2:** Cost breakdown of the prototype