Robitics: A solution for Visual disabilities

By Eric Marquez

Eric Marquez

Dr. Haesun Lee

COSC 4395.001

1 December 2020

Table Of Contents

1. Abstract
2. Introduction
3. Design and Implementation
4. Challenges
5. Pros / Cons
6. Future Enhancements
7. Conclusion
8. References

Abstract

Disabilities have been a persistent problem for mankind. Those who are most unfortunate to be inflicted by these disabilities are bound to live a life full of challenges and inconvenience. In 2016, a survey was conducted by the NFB (National Federation of the Blind), and the results have shown that there are 7,675,600 people that are blind that live in the United States. This is a staggering number that underscores the urgency that should be carried for finding a modern solution to this disability.

The purpose of this study is to present a logical solution to aid those with visual impairments using modern robotics technology. The technical question to investigate is the following: For those that have a visual disability, are modern technologies and sensors an adequate solution to replace current visual aids?

It is hoped that this research can show that modern technologies are ready to be implemented to help those not only with visual impairments, but also those afflicted by other disabilities as well.

Introduction

Technocultural evolution driven by epigenetic and memetic factors, the collective will of humanity to emulate the most successful achievements of Nature with technology, proceeds now millions of times faster than genetic evolution of our species at an ever-accelerating pace. (Alex M. Vikoulov, TECHNOCULTURE: The Rise of Man)

As Vikoulov states, with technology we are trying to emulate tasks accomplished by nature the further we progress as a species. It seems to be such a grandiose idea to do so, but this advancement is what is needed to further evolve technologically. This research is also not exempt from this emulation of nature. In this research, at a basic level, a human eye is being emulated.

To achieve this goal, this project will incorporate a myriad of basic robotics components. In conception, the components include a logic controller, several motors, a breadboard, and a glove to serve as the chassis for the robot.

Design and Implementation:

As stated in the introduction, the initial plan was to use basic robotics components that could work in a flexible environment but also be cost effective at the same time. One of the sub goals of this research is also to show that this technology can be affordable. In grand total, the final cost for the project is under 50 USD. With that being said, the components used for the robot are not necessarily practical for the real world (this will be elaborated on more), but they are within realistic means for a DIY project.

The first component that I would like to cover is the logic controller that was used, the Arduino Nano. The Arduino nano is a simple logic controller that is widely used in various DIY robotics projects. One of the main benefits of the Nano is the size that its name signifies. The Nano is without a doubt one of the smallest logic controllers that Arduino offer. The size of the component was considered ideal for this project since the final version of this project needs to be as compact as possible. The reasoning behind this is to be less cumbersome on the user. The user should not feel as if they are encumbered by using the robot. For reference here is an image of the Arduino Nano:

A picture containing text, electronics, circuit

Description automatically generated

The Arduino Nano suited this project perfectly since it is very good for anything that is DIY, although Arduino is not something that is practical in the industry. If this was an actual product, you would want to get every component of the robot built into the logic controller, except for the motors. One would want everything to be as small and compact as possible.

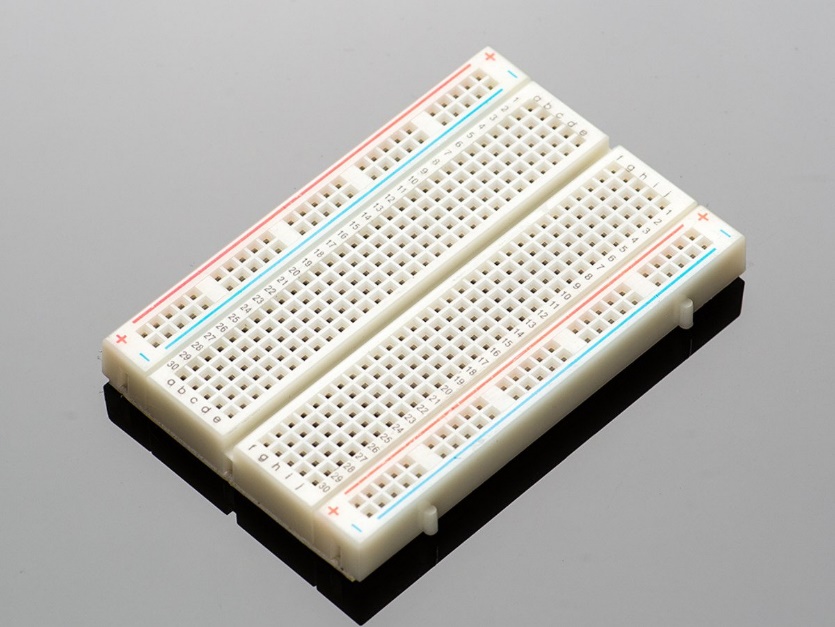
The motor that I used for the project is a standard Servo Motor. Servo Motors have a rotating socket that can rotate a max of 180 degrees. This is easily implemented into the Arduino IDE by using the servo library and using the write function. For reference, here is an image of a standard servo motor:

A picture containing text

Description automatically generated

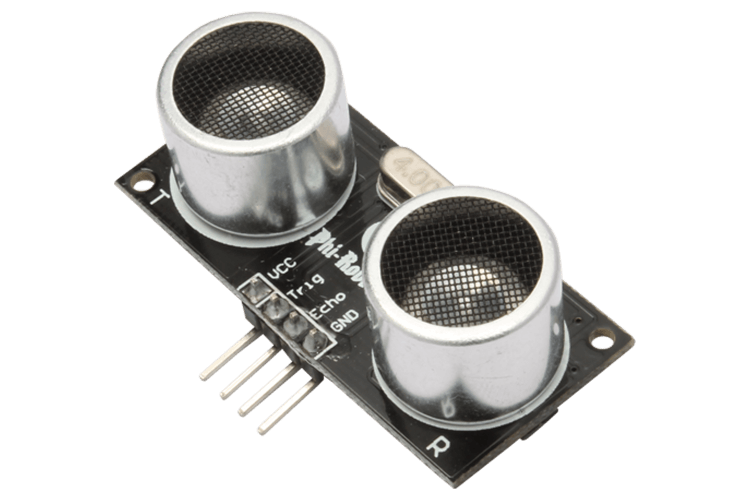
The wires on the servo motor are an industry standard for many components. The red wire is the 5v, the brown is the GND wire, and the orange wire is the PWM. The PWM attaches to one of the digital pins on the Arduino Nano.

Another component that is implemented to the design of the robot is standard breadboard. The breadboard serves as a convenient and innovative way to organize the wires to connect to the Arduino Nano. This is useful because a breadboard allows for multiple components to be powered by the Arduino Nano. This also allows someone to avoid soldering parts due to being able to us simple jumper wires to connect components. Here is a image of a breadboard for reference:



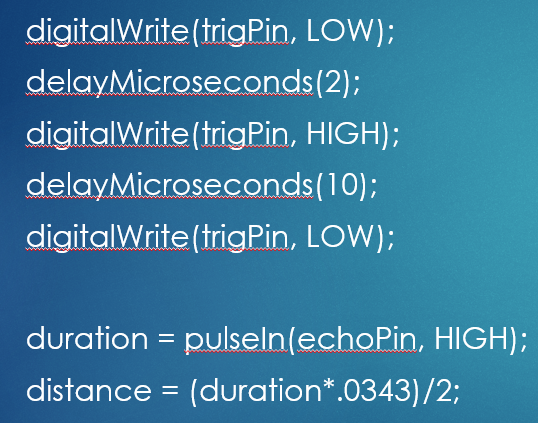
The way you can power multiple components on a breadboard is very simple. The blue strip represents the negative charge for a bread board and allows one to connect the ground pin of an Arduino Nano to this strip. Once done, the entire strip of blue pins act as ground pins. The red strip is very similar, but it represents a positive charge. Like the blue strip, one can connect the 5v pin to the Arduino Nano to this strip, and all pins on the red strip will now be considered 5v pins.

The last major component for this research project is the sensor. Last but certainly not least, the sensor of this robot needed to be something that could measure distance. In the end, sonar was the definitive choice. Sonar has little interference due to the use of high pitch frequencies, and it is also a more affordable technology compared to other sensors. Here is a picture of the sensor:



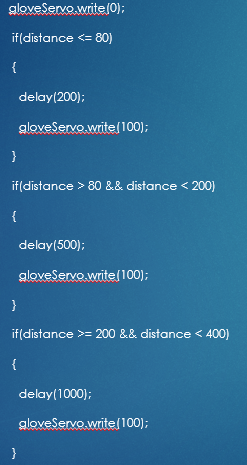
This component would serve as our eye for a robot. The pins on this component are surprisingly simple. The VCC and GND pins need to be connected to their respective power sources on the breadboard (the 5v strip for the VSS, and the GND strip for the GND). The trig and echo pins need to be connected to the digital pins on the Arduino. The way this sonar sensor works is that it sends out a high frequency pitch and waits for this pitch to reverberate of an object. If one can time how long it takes for the sound to reverberate of an object, then one can find the distance by utilizing simple algebra and knowing the speed of sound.

The major logic challenge for the robot is about finding the distance from the sensor to an object. Once this dumber is obtained, one can manipulate the servo motor based on the distance. Here is what the snippets of code look like to accomplish this:



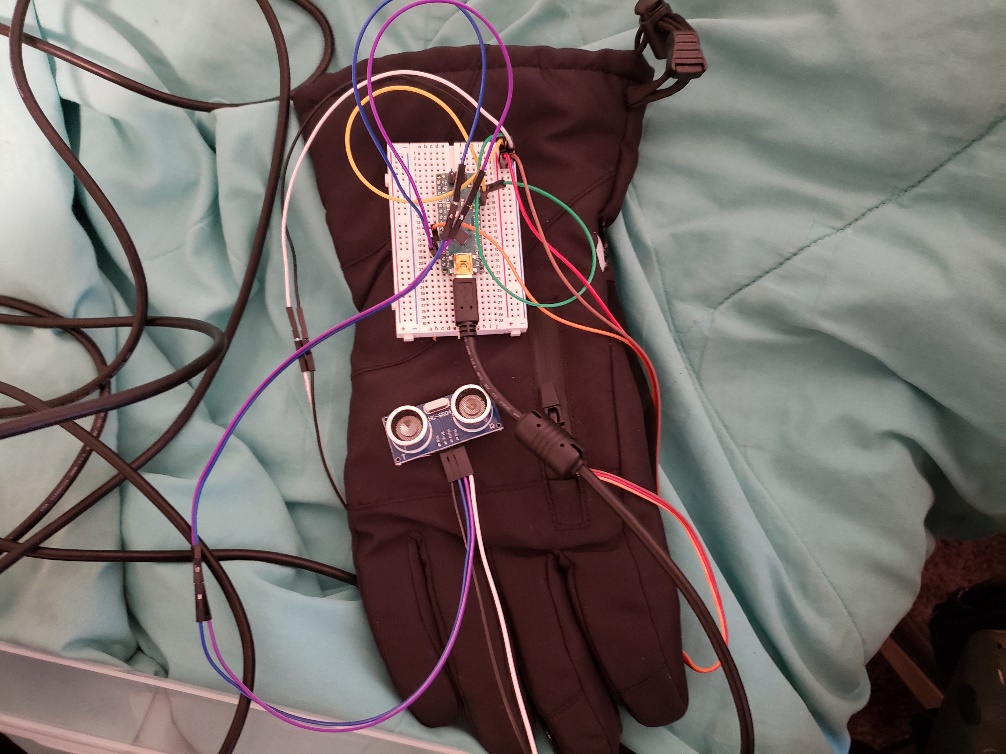
This script in particular is what lets us calculate the distance. First the sensor is set to low for two microseconds to ensure that the trigPin is set to low at the beginning of the loop. Then, we can set the trigPin to high for ten microseconds, which sends out an eight-cycle sonic burst from the sensor. These high pitch frequencies will then bounce of an object and return to the sensor and set the echo pin to high for however long the sound waves were traveling. After this, we need to use the pulse-in function to determine the duration that the pin was set to high. Once we have the duration, all we need to do is use the formula for speed to determine the distance. So we know that speed = distance / time. If we use simple algebra, we can determine that distance = speed \* time. The speed of sound in microseconds (Arduino records time so we cannot use the standard second) is 0.0343. Lastly, to determine the distance, we need to divide by two since the sound wave traveled to and from the object.

The next piece of script is to manipulate the servo motor. Initially, the plan was to have the servo motor apply different points of pressure to tell the user how far an object is from them. However, later it was determined that it would be best to simply use vibration to accomplish this. Here is the code snippet:



This is the main logic to simulate a vibrating pulse for the servo motor. Through trial and error, it was determined that anything less than eighty centimeters would be considered a close distance to an object, anything between eighty and two-hundred centimeters would be considered a medium distance to an object, and lastly anything between two-hundred and four-hundred centimeters is considered far from an object. Through a system of delays in microseconds, the servo motor can rotate back and forth zero degrees to one-hundred degrees with varying speeds. This simulates a vibration and alerts the user to how close an object is. If the servo motor is vibrating fast, then you know an object is close. If it is vibrating slowly, then you know that an object is far. And lastly if the motor is vibrating at a pulse in between these two speeds, then an object is at a medium distance to the user.

The final design of the robot has a glove to adhere the breadboard, Arduino nano, and servo motor. The user can feel the vibration of the servo motor on the top of their hand with this design. Here is the final design of the robot:



Challenges

The challenges of implementing this solution were very unexpected roadblocks. The first challenge came with how I perceived a sonar sensor would work. Initially, I thought that the sensor could determine what direction the objects in front of it were (i.e. an object is to the left of the sensor). Upon realizing that the sensor can only determine the distance of an object in front of it, I had to change my approach. Initially, I wanted three servo motors with varying points of pressure to determine where an object was. One servo motor would determine objects on the left, another servo motor would determine objects to the right of the user, and lastly the third servo would determine the distance of objects in front of the user. This is not possible unless three separate sonar sensors are implemented. I decided to scrap the above idea since I believed three sonar sensors is too cumbersome for the user.

The next challenge I encountered is related to the Arduino Nano and it’s power source. Initially, I imagined this robot being portable by employing a battery sensor. However, until I received the Arduino Nano, I did not realize that it does not have a slot for an AC power connection. There are micro-usb battery packs that you can buy, but upon further research, it is widely considered a bad practice sense this can short out pins on the Arduino Nano. Essentially, I failed to realize that the Arduino Nano is more for stationary projects. Despite this setback, I was able to circumvent this during the demo for the project by using a long micro-usb cable.

Pros / Cons

The pros and cons for this robot are equal in gravity. The pros do not out-weigh the cons and vice-versa.

Pros for this robot are the following:

* The robot is extremely cost efficient and budget friendly.
* The robot is a great proof of concept that it is possible to use sonar to emulate a human eye.
* The robot is surprisingly light-weight and does not incumber the user in any way.

Cons for this robot are the following:

* The robot cannot utilize batteries to be portable.
* The robot is not commercially practical.
* Sonar limitations:
  + Objects need to be mostly parallel with the user.
  + Some objects that do not have a hard surface do not reverberate sound properly.

Future Enhancements

This robot was implemented efficiently, but there are many improvements that can be made. The most crucial enhancement that would greatly improve the performance of the robot would be the implementation of a lidar sensor. Lidar technology is extremely reliable and accurate. Currently Lidar sensors are being used to test self-driving cars. Unfortunately, lidar technology is extremely expensive; however, in the future, lidar technology will most likely fall in price as it ages. This will make the technology more commercially available.

Other enhancements to improve the robot would be to use more practical components for a product. The idea behind this is to appeal to the user as much as possible. Essentially, most users would prefer a product that would seem like they are not using a visual aid at all. To accomplish this, one would need to use as small of components as possible to hide them in a glove. A more specific suggestion, for example, would be to use the same vibrating motors that are used in smartphones. These motors are much lighter and more compact. Perhaps this solution could be implemented into a smartphone app that uses the user’s camera on their phone to act as their eyes. This would be extremely convenient since vibrating motors already exist in phones.

Conclusion

In closing, this robot underscores how current technologies are ready to aid those with disabilities. Cybernetics were once considered something that is only seen in sci-fi blockbusters, but now they are seeming more and more real-world applications. Despite this, the subject of using robotics for these endeavors is considered by some to be understudied. In this quote by Andrea M. Kuczynsk, she states the following, “Robotic technology offers the potential to objectively measure complex sensorimotor function but has been understudied in perinatal stroke” (1). Like Kuczynsk states, robotics is understudied in aiding those with cerebral palsy. Not only for this specific field, but one could argue that robotics is understudied with disabilities in general. From cybernetic prosthetics to sensory aids, robotics can aid those that are less fortunate. All that is needed is a will to innovate and a desire to help our fellow men and women.

Works Cited

Kuczynski, Andrea M., et al. “Kinesthetic Deficits after Perinatal Stroke: Robotic Measurement in Hemiparetic Children.” Journal of NeuroEngineering & Rehabilitation (JNER), vol. 14, no. 1, Feb. 2017, pp. 1–14. EBSCOhost, doi:10.1186/s12984-017-0221-6.

Vikoulov, Alex. TECHNOCULTURE: The Rise of Man (The Cybernetic Theory of Mind). San Francisco CA, Ecstadelic Media Group, 9 September 2020.

“Getting Started with the HC-SR04 Ultrasonic sensor.” Arduino, 9 August 2017, https://create.arduino.cc/projecthub/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-036380.