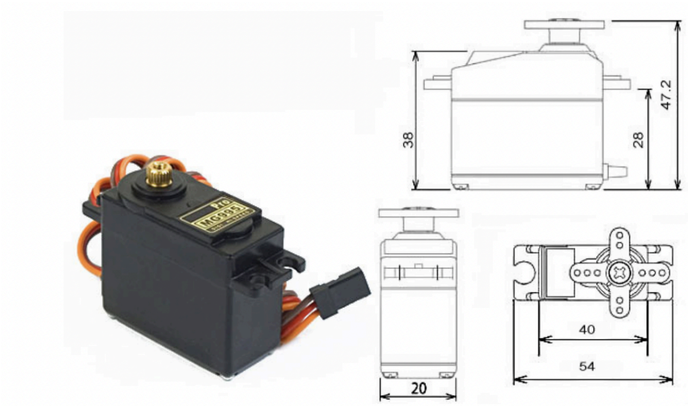
LiDAR Room Scanner Project
Eric Sander
4/30/2025

Introduction:

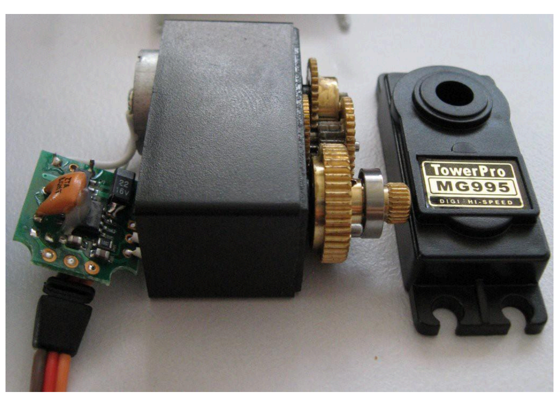
My goal for this project was to create a lidar room scanner that could move with the use of a servo and a stepper motor to make 3D models of rooms on a computer over serial connection. The LiDAR system is modular so that the user may make adjustments to the range and resolution of the scan from the camera without much work needing to be done. The final design uses the computer for user display and progress, however progress is relatively obvious from the direction of the camera. The broader reasoning for why more effort was not made for a user interface element is because other projects online of the same nature already included such elements, which makes it not a meaningful or novel addition to the final product. The unique feature of this project is its strategy to reduce the scan time of the camera system with servos by modding the servo with a soldered wire on the internal potentiometer of the servo. While the servo moves back and forth at max speed the angle and distance are sampled at a 500Hz frequency. Receives ~158 samples per vertical movement and 180 samples along the theta spherical axis. This leads to a total of 28,440 samples which is enough for satisfying results with the scan. Compared to a similar project I found online, I am able to collect more than twice the samples in half the time, meaning this project is more than four times faster than a naive approach to this problem.

Components:

I will now walk through the main components that I used for the project. The first is the servo that is the key for the sped up feature of this project.



Datasheet link: [Servo Datasheet]



The above photo is not one that was taken by myself, but it shows the gearbox and the potentiometer inside of the MG995. A wire that captured the analog signal on this wire was soldered into the servo and input to the arduino for the ADC to analyze and convert to the value of the potentiometer to get the angle of the servo at the moment of collection.

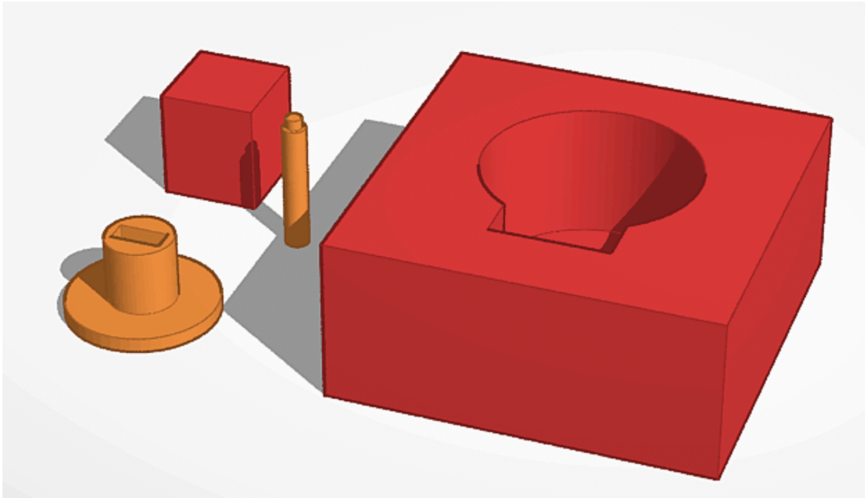
The next component is the LiDAR module that I purchased off of amazon. It is the TFMini-S camera and it comes with an arduino library that I tested with the camera to make sure that it works, and it performed as expected. The camera could push up to a sample rate of 1000Hz, but I found good results with half of that frequency. Ideally the sampling freq should be adjusted for the distance that you are planning on scanning and the desired resolution. Because I did plotting with the matplotlib library, the point clouds are good enough to get a clear scan of the room, but other means could be utilized to get a better representation of the points fed to the computer by the arduino. Because of this having more points did not always make the resulting image more clear. The library for the TFMini-S was TFMPlus. Additionally, I used libraries for the servo and stepper motor.

When I first received the LiDAR camera, I performed a quality check by just streaming the distances that it picked up with the library, and I found that the quality of the scan and the limited noise for large distance changes was pretty high quality for the price.

TFMini-S Datasheet

Finally I used the standard stepper motor given to us in our arduino kits for changes in the theta direction.

Using the dimensions of the stepper motor I printed the following case for thew stepper motor. This is one of the components that I designed myself, as well as the cylindrical attachment that connects to the stepper motor shaft and holds the LiDAR pan tilt up. This stabilizes the camera system as I found during early testing that it would wobble and turn a lot white the servo was moving. The other two components were printed when I was using a smaller servo component. I later switched to a larger servo for convenience reasons as the pan tilt that I printed from an online source was designed for a MG995 or equivalently sized servo.



Lastly, the box was designed and printed for the pan tilt camera to stand on and to hide the messy wires present in the system. This was designed in illustrator, however the puzzle piece features of the box were heavily inspired by how I saw my classmates printing boxes.

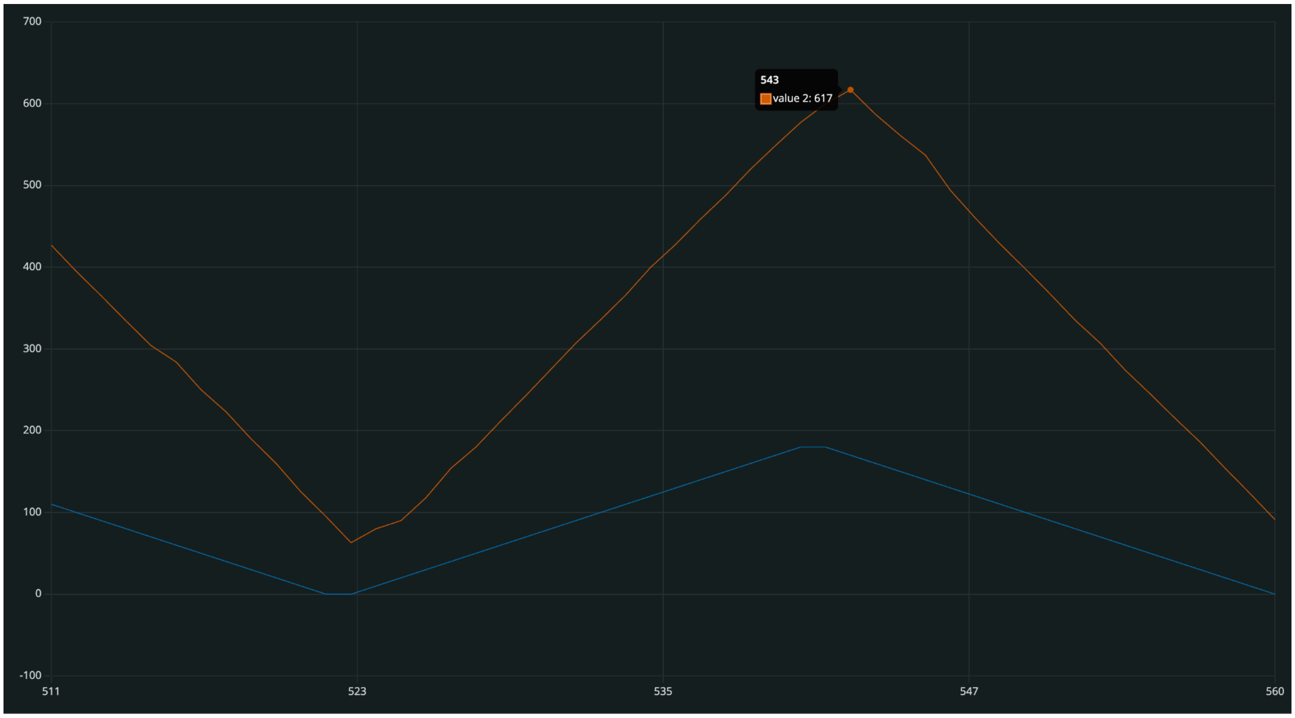
Design Process:

The LiDAR camera:

The first step of the design process was to ensure that my LiDAR camera was suited to the task and could accurately scan distances at a rate above 100Hz. I created a quick sketch that used code from the library and manufacturer to test the LiDAR camera and understand how to work with the library and get the result I need. At first, the distances were streamed to my computer, however, doing this was difficult to allow for accuracy in terms of matching distances with the appropriate angles at which they were calculated, but I found that I could just poll the camera and get the result from the camera quickly as well and polling it made it much easier to get the correct position of the camera while it was scanning. Other than this change, the camera logic and implementation described in this section is how it worked in the final design of the project.

Reading from the potentiometer:

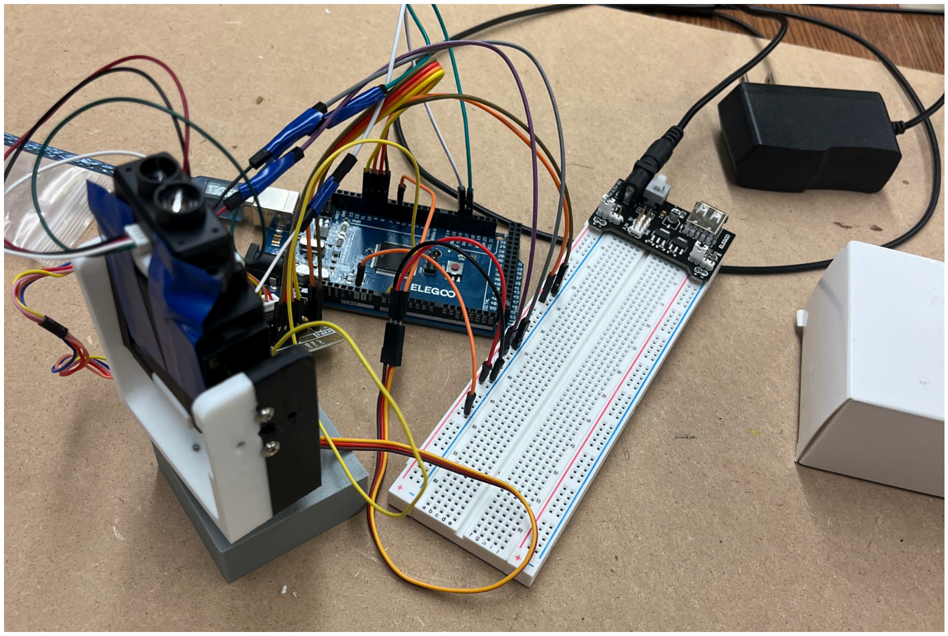
The next part was admittedly a little frustrating, and it was attaching the potentiometer to the servo motor. I did this once with the mini servo and the MG995 servo I used in the final implementation, and they both had consistent results. Both had slippage when changing directions, and I planned to counter this by making the servo turn further than it needed to and adding a delay during the sweep to collect data when the change in angle was very smooth, but implementation without this feature still yielded effectrive results. Below is a diagram of the angle of the servo and the potentiometer value. Note they do not line up perfectly as the servo takes time to arrive at the position it is moving to, but in the project I read directly from the potentiometer so this was not a problem.



I also played with the idea of adding an RC circuit to the potentiometer wire to reduce the noise but results shown in the graph were satisfactory. The graph above is also made with the strategy of moving the servo from the bottom point to stop point at max speed. This was the final approach I arrived at after a lot of trial and error. Initially, I tried to have the servo make many small and exact steps and get the LiDAR readings along with these, however, not only did that lead to more noise because of slippage, but it was significantly slower than just moving the servo to one extreme once and reading directly from the soldered wire to the internal potentiometer.

Stepper Motor Movement:

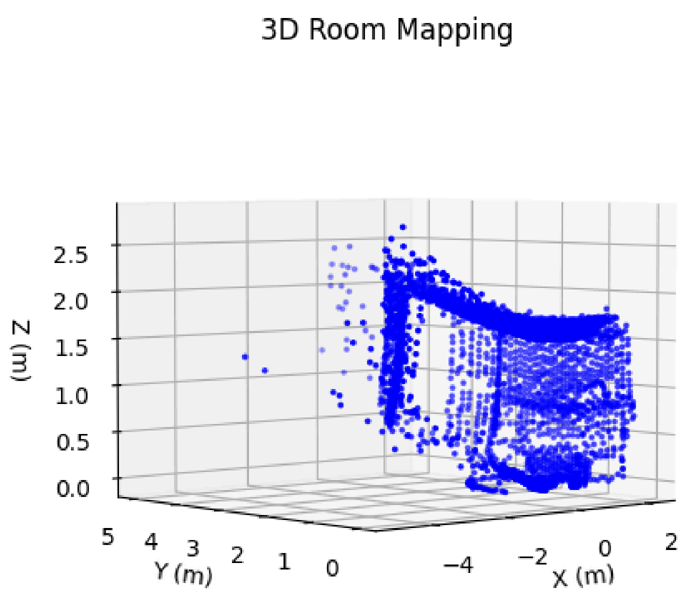
Finally the stepper motor did not cause me as much headache because it did not need to be moved very often and it is pretty exact in its movements. I did find that the motor had a little trouble moving with the load that I had on it and so I needed to find the amount of steps to get the desired angle range I needed. I was able to measure out the steps (written in the code) for a 180 degree turn with a step size of 1 degree each time the stepper motor moved, which got me pretty good results.



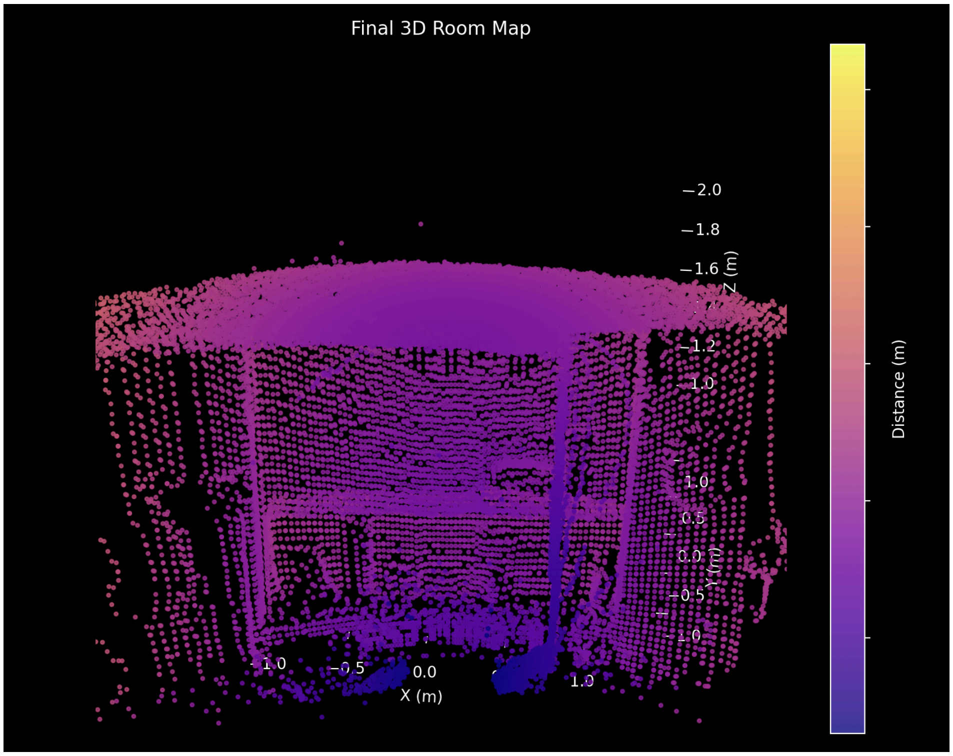
This is what the design looked like after it was working for the first time

Some early problems:

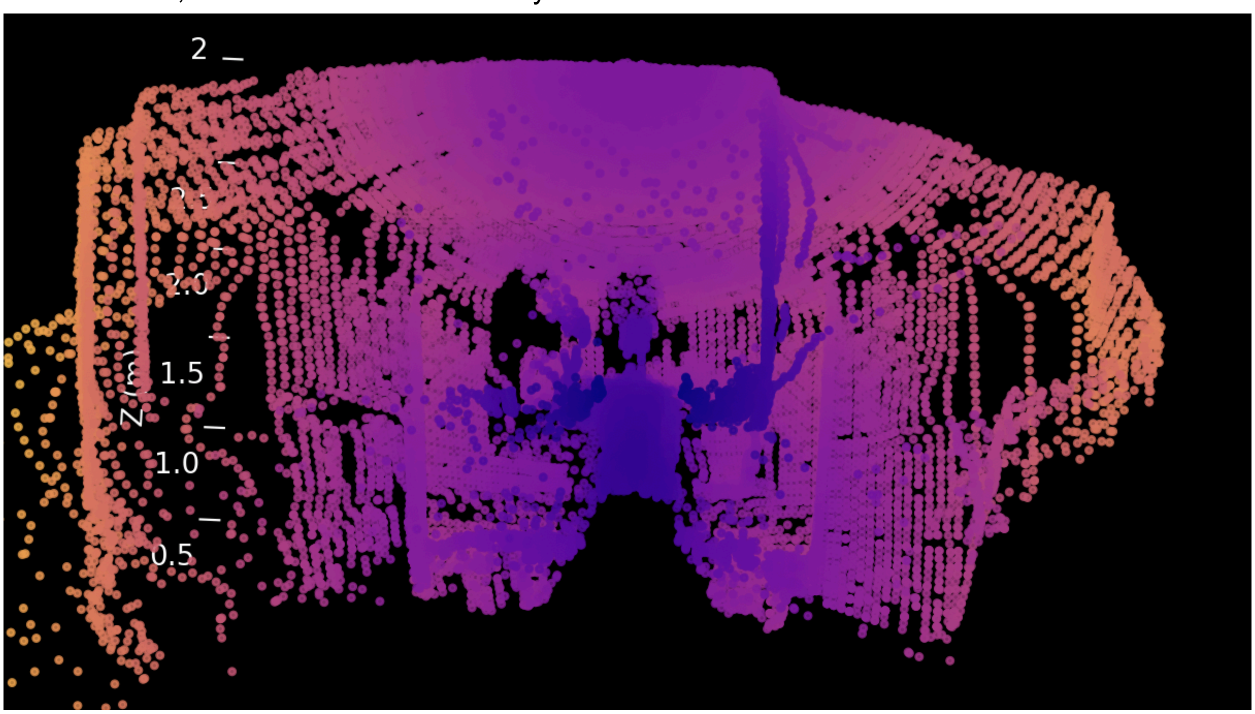
Some issues that I encountered were more mechanical. Because the base of the stepper motor was small compared to the actual body of the pan tilt camera and the wires were stiff, the camera would tilt in an awkward way and cause shakiness in the straight lines of the 3D model.

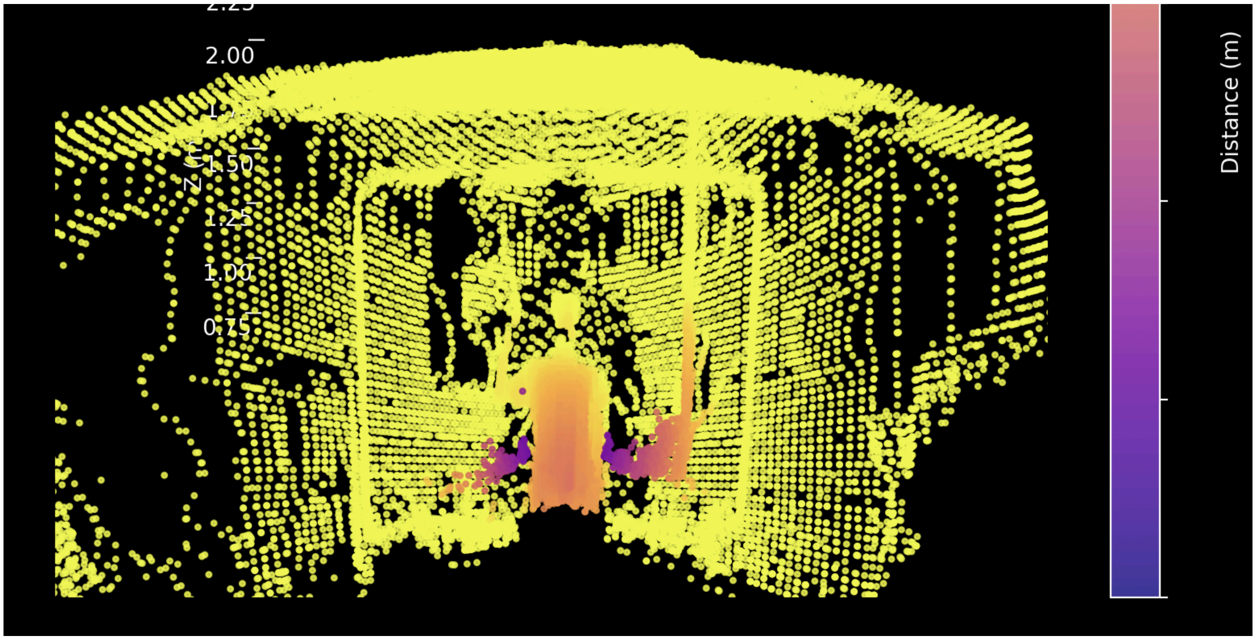


This image shows the instability previously mentioned. Additionally I had an issue with how I was converting to spherical coordinates. I set the 0 degree marker for phi on the xy plane (when converted to cartesian) which made the conversion from spherical captured by the arduino to rectangular bend in awkward ways. After fixing this issue the point clouds began looking a lot cleaner. I made adjustments to add color contrast to make depths and shapes easier to see. After the servo stability fix, points coming from the arduino were very good and surpassed expectations. I made some additional tweaks to the computer display and ended up with graphs that looked cleaner, like this:

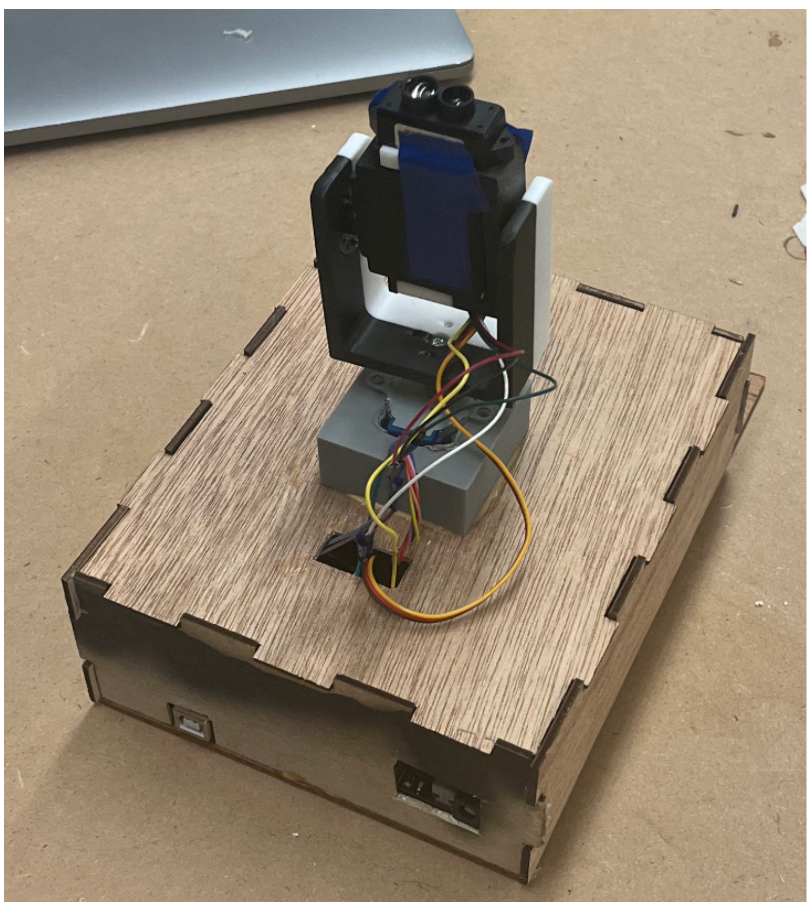


For a little fun, I also made a scan with my hand raised in front of the camera.





The angle makes it a little awkward but my silhouette made it in the image and I was pleased with the results of the camera. The final design with all the bells and whistles is below and worked within expectations. I was very pleased that I was able to achieve what I set out to accomplish.



Sources:

https://www.qcontinuum.org/lidar-scanner

https://how2electronics.com/how-to-use-tfmini-s-lidar-distance-sensor-with-arduino/

https://www.instructables.com/Project-Lighthouse-360-Mini-Arduino-LiDAR/

https://www.diyengineers.com/2022/06/18/lidar-scanner-project-arduino-project-nodemcu/