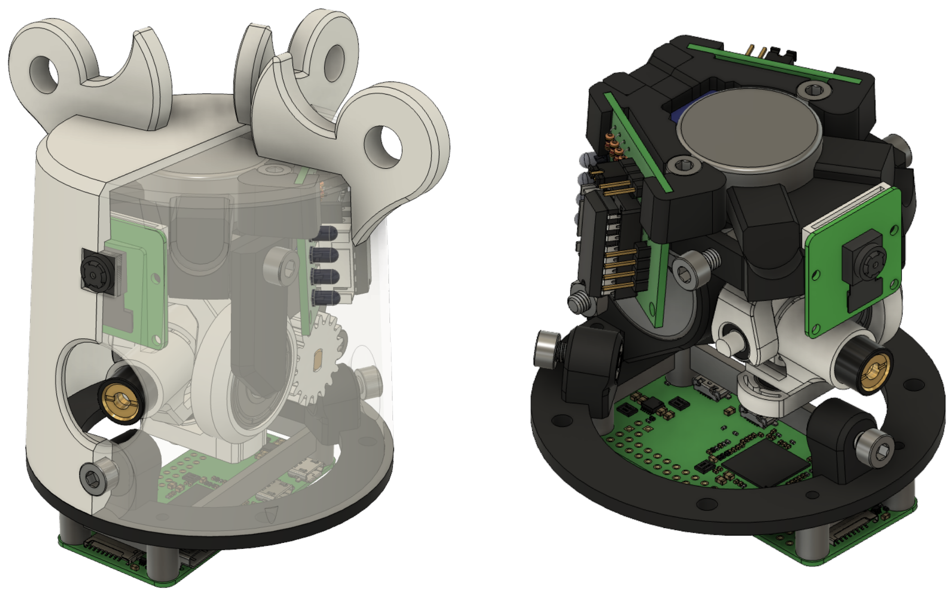
The Guardian: Collector's Edition
Targeting Subsystem Technical Documentation

Hayato Kato
ASME of UCLA – X1 Robotics

2019 - 2020



Abstract

This was a club project that I worked on during my 2019-2020 school year at UCLA. We lacked sophisticated documentation of the entire project, so I have written down the technical documentation of the parts that I was directly involved with, which includes the targeting subsystem. The project as a whole was a follow-up project from a previous year, so many parts of the documentation focus on the improvements my subsystem and I made from the previous generation targeting system which had many issues and design flaws. Overall, the final product sticks to the initial objective of improving the subsystem and works as intended at the end.

1 Introduction

Founded in 2015, X1 Robotics is one of the University of California, Los Angeles (UCLA) Engineering's student-led project teams with the mission of spreading real world engineering and problem solving skills through the development of challenging robotic systems. Falling under the umbrella organization of the American Society of Mechanical Engineers (ASME) UCLA Student Branch, what sets us apart is our passion and creativity; every year marks the start of a new project decided upon by the diverse team of student engineers, and a new cycle of research, development, and iteration. X1 is proud of the collaborative work environment it fosters and will continue to excel in the field of robotics as it attempts complex, challenging projects pushing UCLA students to their limits. This year, X1 Robotics' resources were divided between two teams. Overall, the club utilized 2 project leads, 6 subsystem leads, 2 subsystem mentors, and 54 total members. Applications were sent out at the beginning of the year, and the team was assembled based on prior knowledge, diversity of experience, and willingness to commit through completion. A full list of the members responsible for completion of The Guardian: Collector's Edition project is as follows:

- X1 Project Lead
  - John Tabakian (3rd Year Mechanical Engineering Major)
  - Akaash Venkat (4th Year Computer Science Major)
- Chassis Subsystem
  - Andrew Miller (3rd Year Mechanical Engineering Major)
  - Linda Zaragoza (3rd Year Mechanical Engineering Major)
  - Reilly Terao (2nd Year Mechanical Engineering Major)
- Legs Subsystem
  - Jingbin Huang (4th Year Electrical Engineering Major)
  - Sruti Munagala (2nd Year Mechanical Engineering Major)
  - Daniel Nashed (4th Year Mechanical Engineering Major)
  - Kyle Ignacio (4th Year Mechanical Engineering Major)
  - Nicholas Zhao (2nd Year Mechanical Engineering Major)
  - Phillip Ko (2nd Year Mechanical Engineering Major)
- Targeting Subsystem
  - Hayato Kato (3rd Year Electrical Engineering Major)
  - Vishal Kackar (2nd Year Mechanical Engineering Major)
  - Zack Yuan (2nd Year Mechanical Engineering Major)
- Software Subsystem
  - Sangjoon Lee (3rd Year Computer Science Major)
  - Audi Liu (3rd Year Mathematics of Computation Major)
  - Bradley Pickard (2nd Year Computer Science and Engineering Major)
  - Anthony Tate (2nd Year Computer Science Major)

2 Inspiration

The Guardian project is inspired by Nintendo Corporation's game called The Legend of Zelda: Breath of the Wild. In the video game, the player encounters an enemy character known as a Guardian: a six-legged mechatronic creature with a free-rotating head and a laser-shooting eye. The creature roams around in an open field before identifying the player, rapidly approaching them, and then attacking them with its laser. X1's Guardian attempts to mimic its in-game counterpart in aesthetics, functionality, and behavior.

We picked this project idea last year during the 2018-2019 school year and brought the project to completion by the end of the year. However, the club as a whole found many points of improvement across the entire project, and unanimously decided to continue working on the project for a second year to improve on the design. Having worked on a different subsystem previously, I was tasked to become the targeting subsystem lead for this year to fix some of the flaws that made this particular subsystem fragile and unreliable.

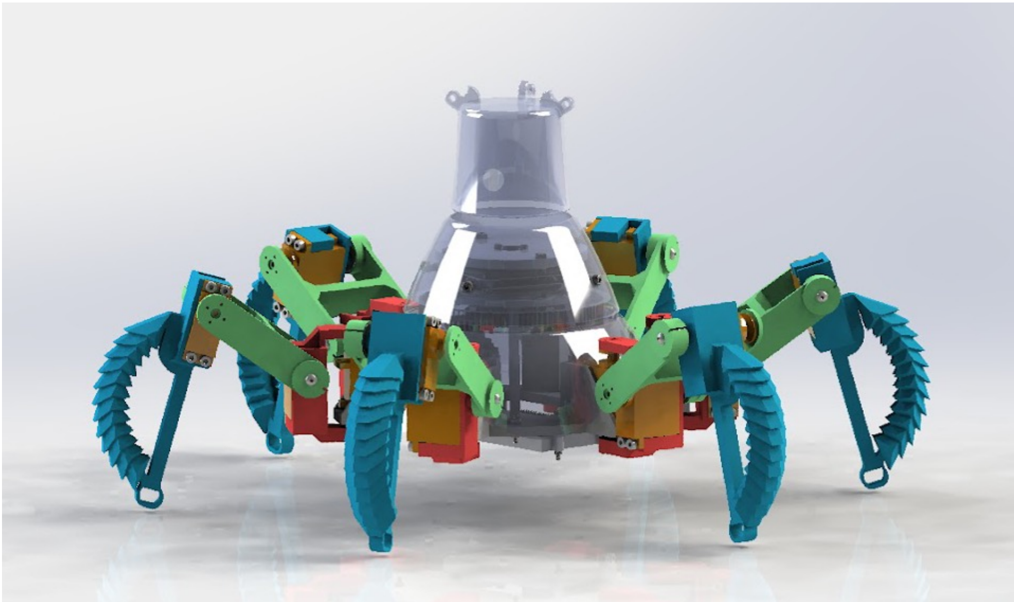


Figure 1: Guardian Hexapod Robot Full Assembly Diagram

3 Overall System Breakdown

3.1 Chassis Subsystem

The chassis subsystem was responsible for designing the main body of the hexapod robot which holds the Raspberry pi and the servo hat boards used to control all of the servo motors used to actuate the 6 legs with 3 degrees of freedom. Design considerations included special focus on accessibility to the internal electronics and lightweight components to reduce load on each of the legs.

3.2 Legs Subsystem

The legs subsystem was responsible for designing a rigid 3 degree of freedom leg which would allow for a strong yet agile leg movement required for making the hexapod walk. Design considerations included the ease of assembly considering how 6 of these legs needed to be continuously assembled and maintained, alongside ensuring cost efficiency. The aesthetics of the legs were also taken into account to keep true with the in-game visuals.

3.3 Targeting Subsystem

The targeting subsystem was the subsystem I was responsible for, which had the objective of creating a laser targeting system that was capable of pointing the laser to a desired position within the field of view of a mounted Raspberry pi camera. One of the main considerations was the accuracy of the laser pointing mechanism and the size of the system, which needed to fit within the volume of the shell of the Guardian's head model.

3.4 Software Subsystem

The software subsystem was responsible for creating a web-interface used to communicate with the hexapod so that it can be controlled remotely, alongside developing an image detection system to allow the laser-targeting system to discover a target within view and give appropriate commands to the targeting system regarding where in view the target is.

4 Preliminary Design Brainstorming

4.1 Motivation

The main objective of the targeting subsystem was to design and produce a laser turret system that could accurately point the laser diode towards a target detected by the vision AI subsystem. This mimics the in-game character's attacking motion when it makes sight of the enemy and shoots a beam of energy after locking onto its target. The mechanical subsystem from last year designed and created a full pan and tilt mechanism that was incorporated into the full assembly. Despite their success during testing, we found many points of improvements that limited its performance:

Last year's system used a 3D printed ball bearing that was controlled using two micro servos mounted at an angle of 90 degrees from each other. The use of additive manufacturing to produce a spherical component that slid inside other plastic parts created a lot of resistance and slack in the mechanism. The use of cheap hobby servos also limited the laser motion's accuracy even at close ranges, since they had a minimum resolution of around 3 degrees. The lack of actual bearings also compromised the mechanical system's robustness and reliability to reproduce the same motion every time. A CAD diagram and picture of the previous targeting system is shown in Figure 2.

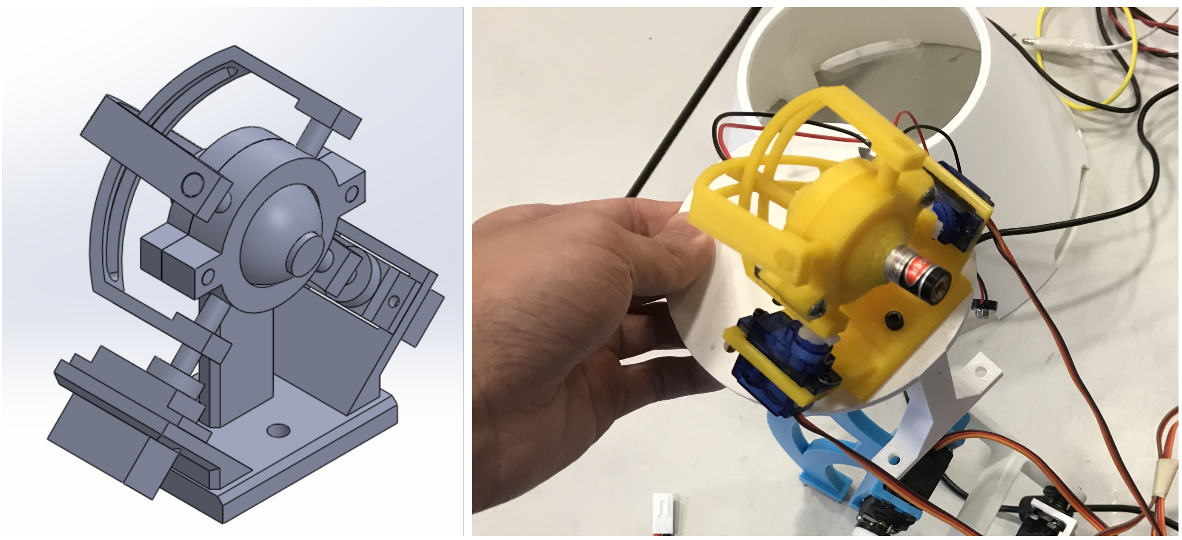


Figure 2: Old Targeting System

4.2 Improved Design Choice

We really wanted to improve on the system's overall accuracy when tracking an object, so we spent the majority of fall quarter researching existing laser pointing mechanisms that are used in the world. We discovered two main methods: the laser galvanometer system and the Agile Eye mechanism, as shown in Figure 3. The laser galvanometer system reflects the laser beam towards a desired direction by tilting a pair of mirrors controlled by two stepper motors. The small mass of the mirrors itself allows for very rapid movement, commonly used in laser projection shows. Meanwhile, the Agile Eye mechanism is a type of parallel mechanism that uses two actuators to independently control the yaw and pitch of the laser itself. It is commonly used in rapidly orienting a camera in a gimbal. After considering the accuracy of each system and the space constraint that we will face when putting this system inside the head's shell, we decided that we will use the 2-DOF Agile Eye mechanism. We also wanted to try implementing the Agile Eye mechanism to try and challenge ourselves with a more complicated mechanical system.

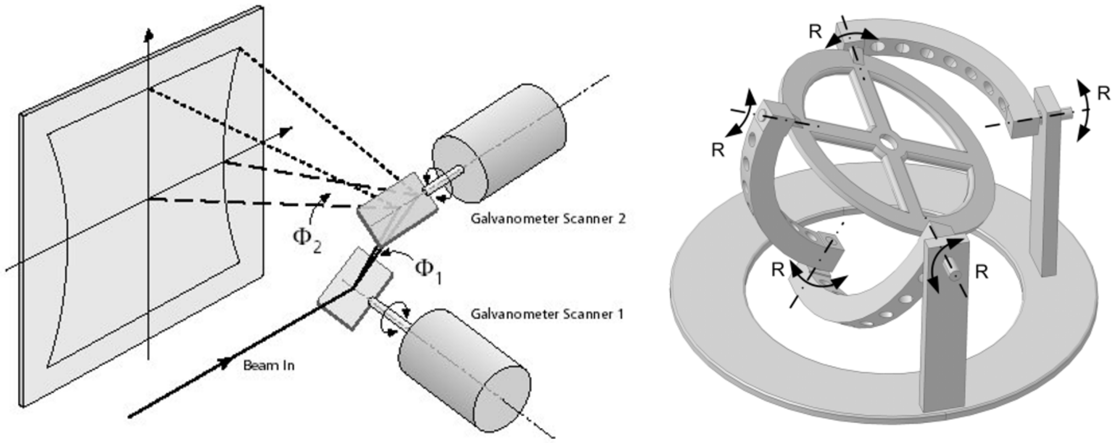


Figure 3: Laser Galvanometer System (left) and the Agile Eye Parallel Mechanism (right)

4.3 Actuator Selection

To avoid the same mistake that we made from last year, we decided that we will use a geared stepper motor instead of the micro servos. This decision mainly came from putting emphasis on the accuracy of the actuator at small angles, which the stepper motor excels at. Looking into available stepper motors in the market, we found the 28BYJ-48 stepper motors that have a decent step count per full revolution (4076 steps to be exact) and a small package size suitable for fitting it inside the shell. A cheap unit price of around $2 also made this motor suitable for this project. To control these unipolar stepper motors, we also used the ULN2003 stepper controller board for easy control using the Raspberry Pi Zero. The stepper motor selected is shown in Figure 4.

Figure 4: 28BYJ-48 Stepper Motor and ULN2003 Stepper Controller Board

The use of geared DC motors and an encoder was also considered, but was ultimately turned down due to the high cost of the motors. However, future development of this project may look into the combination of DC motors and encoders so that a PID control system may be used for smoother motion tracking. One of the limitations of using a stepper motor was its physical limitation of only having discrete angular positions that the motor can be positioned in. The use of geared stepper motors help minimize these individual steps to make it seem like a smooth motion, yet it is still ultimately individual steps with small delays in between. This improvement can help make the system even more accurate and more agile if this functionality is needed in the future.

4.4 Potentiometer

One of the downsides of using a stepper motor is that it is an open-loop system, where the microcontroller has no idea whether the angles specified by the program were actually achieved. The stepper motors could have failed due to external forces preventing the rotation, and the microcontroller will have no idea where the motor is currently pointed towards. This problem also occurs at bootup, since there is no "zero position" that the program can specify to center the laser if it is moved when powered down. These factors made us consider the use of absolute encoders that could monitor the angle of the stepper motor independently. This way, the microcontroller is able to reference the motor angle whenever it needs to and adjust accordingly. It is true that the angular accuracy that comes from the stepper motor itself and the potentiometer readings are redundant, but it was ultimately decided that the complexity of the mechanical system would make the stepper motors more prone to failing. We later found out that due to the consequence of the mechanical design, the yaw and pitch control of each axis gets skewed at extreme angles, thus the potentiometers can be used to help compensate for those errors. The potentiometers that were chosen were SMD components that were meant to be soldered onto a PCB board, thus having a very slim profile that was perfect for our application.

5 Research and Development

After brainstorming and figuring out the mechanical system that will be used, much of winter quarter was spent creating a prototype that allows us to play around with the mechanism. A 3D-printed prototype was modeled referencing a video of an Agile Eye mechanism built by Laval University Robotics Laboratory (`https://www.youtube.com/watch?v=NsGzLuLJWFQ`). This video shows two DC motors used to control each axis of rotation independently to track a reflection marker mounted on the end of a stick. This demo perfectly represented the end product that we wish to make, thus we closely mimicked its design and played around with it to understand its properties.

5.1 Code

The code that controls this system needs to interface between the vision team's code and the targeting system. The code that tracks the target and the laser pointer gives coordinates in pixel values, while the targeting system works in potentiometer values and stepper counts. To solve this problem, the Raspberry Pi's camera FoV was determined, the FoV was converted into pixel values, and the pixel values were converted into stepper counts. Since the vision code actively tracks where the laser pointer is, the conversion described previously is meant to keep the targeting system moving in the correct direction if the laser coordinates are unknown.

The coordinates of the laser pointer and target are not always known, however, so the targeting code must be smart enough to continue looking for the target based on its most recently known position. The code keeps track of the previously known coordinates. If the current and previous coordinates of the target are unknown, then the system doesn't move. If a new set of valid coordinates is sent, then the system starts moving towards the new coordinates and stores them as the previous coordinates. This way if the vision system loses track of the target's location, the targeting system can use the previous coordinates as the destination.

Since the targeting system runs faster than the vision system, a for loop is used to move the laser pointer in between coordinate updates. The code calculates a rough number of steps needed to take in order to get from its current location to the target's location and iterates with that as its upper bound. As soon as the coordinates of the target are updated, even if the for loop is still running, the for loop is terminated and a new one is started with the new coordinates. A manual delay is added to the for loop to help stabilize the targeting system's code.

5.2 Prototype

As referenced above, our initial designs were based on the 2 DoF Agile Eye mechanism created by Laval University Robotics Laboratory. This design utilized two separate “arms” freely controlled by each stepper motor. The steppers were able to move independently and perpendicular of each other by rotating the mechanism within a set of small bearings. This initial design was primarily focused on a proof-of-concept; as such, it was bulky and made no attempts to consider size/weight constraints. However, this prototype worked very well and allowed us to test various programs early on in the project. This design was further refined later in the year as we integrated with the chassis team, as shown in Figure 5 by the second version of the prototype that compacts the laser holding gimbal mechanism while experimenting with the stepper motor placements.

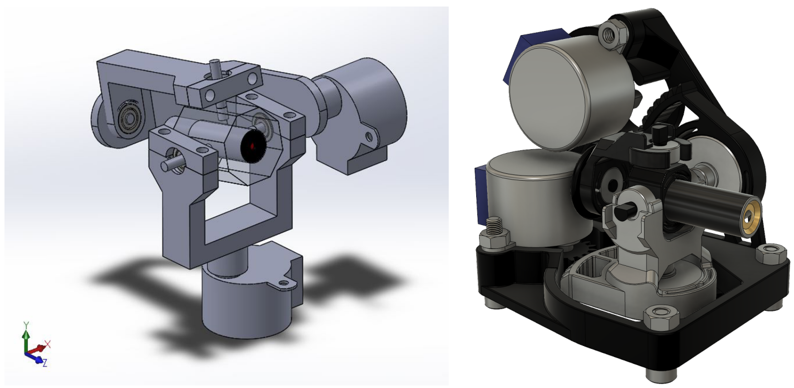


Figure 5: Targeting System Prototype V1 (left) and V2 (right)

6 Final Design

The redesigned CAD render of the new targeting system is shown in Figure 6. This design was derived directly from the original prototype that was made during winter quarter, where each component is compacted down to reduce size while improving printability. Here, the white parts are the parallel targeting mechanism that enable the laser to be oriented in a desired way, and the black parts are the structural pieces that hold the different components within the shell. A stepper motor is placed on top which enables yaw rotations directly by moving the Y-shaped laser mount side by side, while another stepper motor in the back transfers rotational motion through the spur gear and down to the arm that is supporting the laser from below. Micro bearings are used at each joint to enable smooth operation, and each piece is designed with the assumption that they are going to be 3d printed in advance.

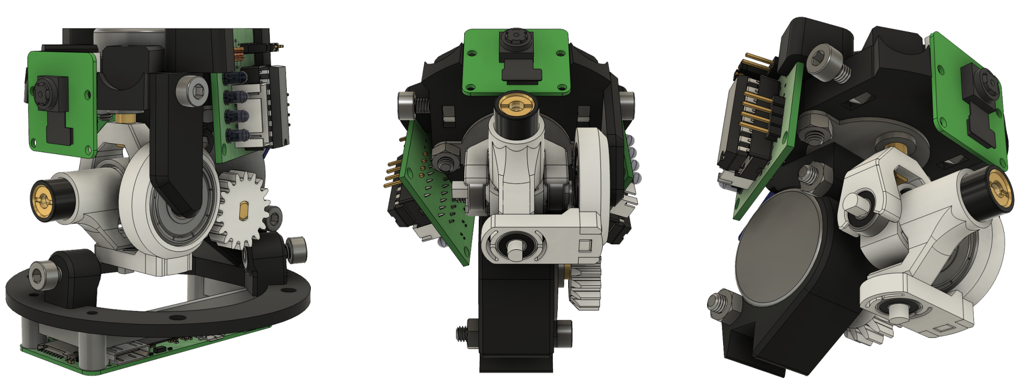


Figure 6: New Targeting System Design from Various Angles

The outer shell of the targeting system was designed by using an image of the character from in-game as reference, as shown in Figure 7. Here, a circular window is opened in the middle to allow for the laser to move around, and another hole is opened on top to place the Raspberry Pi camera. Additional holes are also placed around the shell to easily attach and remove the shell from the internal components simply by removing these bolts. A sectional analysis of the targeting system shows how closely the internal components are packed within the shell. The range of motion of the laser is approximately ± 12 degrees in both the pitch and yaw directions, with a theoretical accuracy of approximately 0.09 degrees per step.

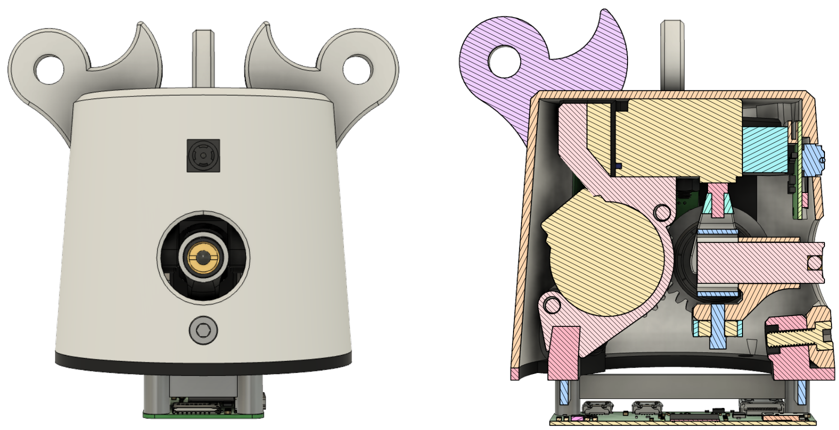


Figure 7: Outer Shell Design and Sectional Analysis of Targeting System

Additional pictures of the assembled targeting system are shown below, where the wire for the electronics are also soldered directly onto the boards. The connection to the Raspberry Pi are established through jumper cables that are passed through the side of the mechanism on the left side, which include the power lines for the stepper motor, the laser, and the potentiometer wires. The size of the overall system can also be put into perspective with how small the shell is compared to a hand and the pliers that are shown in the background.

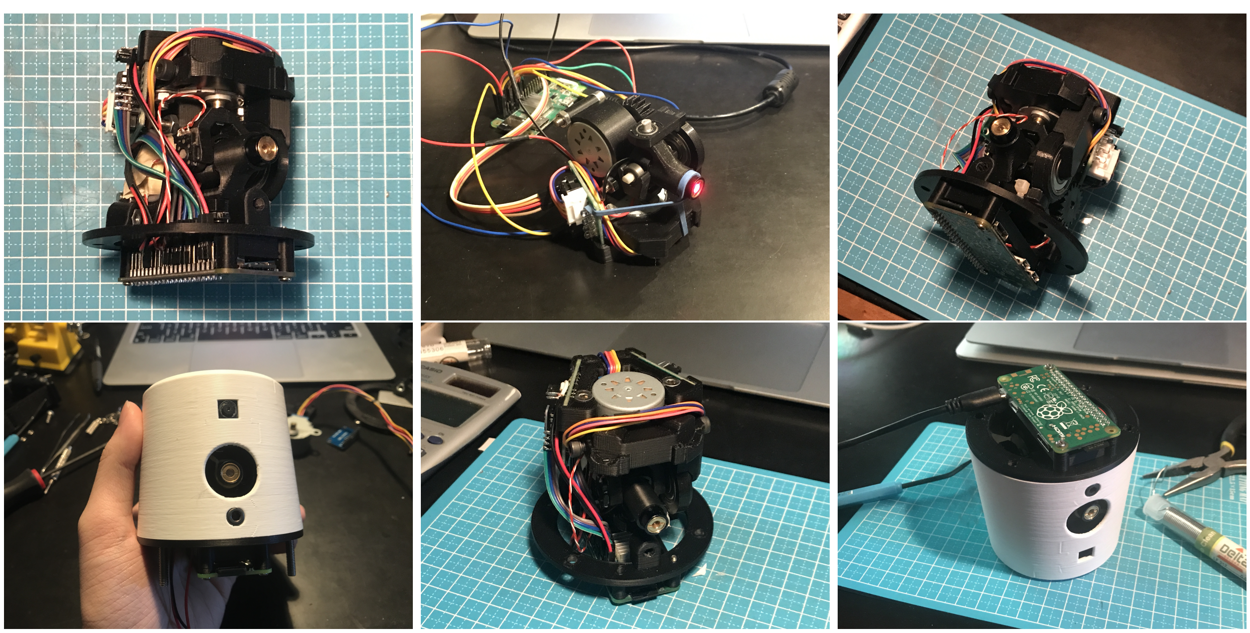


Figure 8: Assembly Process of the Targeting System

7 Results

Although we were unable to complete extensive experiments that integrate the image-detection system and object tracking features (mostly due to a lack of in-person time because of the pandemic), our subsysteam was able to create several demo patterns that were hard-coded into the onboard Raspberry Pi to draw out specific shapes such as a circle and a star. From preliminary tests, we determined that our hardware was stable enough to handle any commands from the software team, and the main bottleneck of the entire targeting system shifted over to the slow speed of the onboard image recognition done my the software team. Overall, this project was a massive success in terms of improving accuracy and hardware reliability compared to the previous year's system.

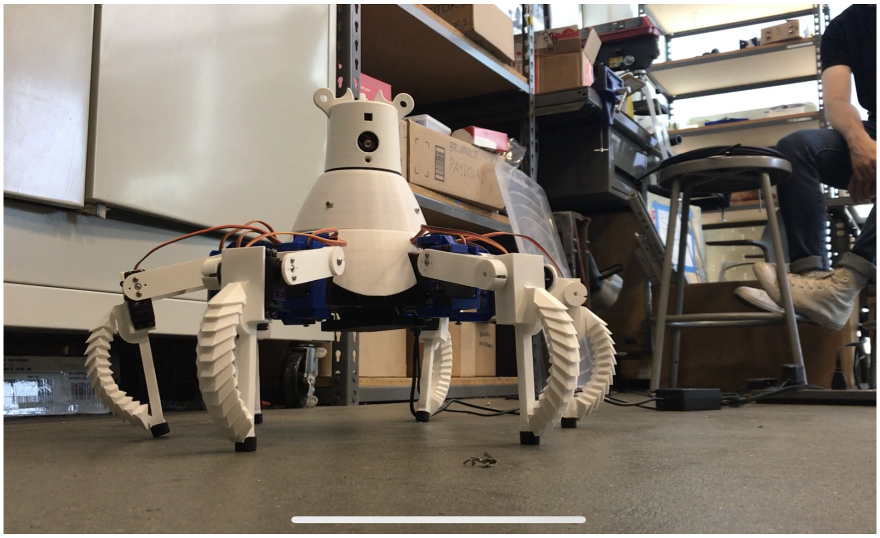


Figure 9: Finalized Guardian Robot in Operation

8 Appendix

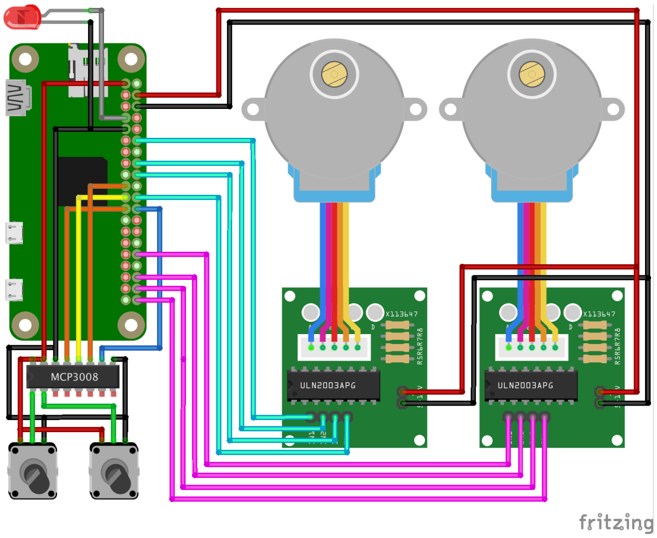


Figure 10: Targeting System Circuit Schematic