ME 401: Mechatronics Lab

Midterm Project: Zombie Apocalypse

Titus Battisti, & Rafael Romero

Section 01L

Data Collected 10/12//2021

Report Submitted 11/3//2021

**ABSTRACT:**

The purpose of this project is to make a robot that can follow different functionalities while being able to change its behavior based on sensor information via radio. The robot must be able to do the following:

1. Use overhead camera information via radio to avoid zombies when in healthy mode.
2. Find and collect balls on the play field to heal the zombies.
3. Switch between states based on data from the radio.
4. In zombie mode use local IR sensors to hunt healthy zombies.

The robot will follow these set rules to play a Zombie Apocalypse Tag. The game starts with three healthy robots on the play field along with a zombie. The objectives of the healthy robots are to catch a healing ball and tag a zombie to heal it; while the zombie’s objective is to catch a healthy robot to convert it into a zombie. While in healthy/healing mode the robots can communicate with the radio which tells the robot the specific location of other healthy robots and zombies. In the play field there is a corner that turns a robot into a zombie mode or healthy mode depending on the corner it touches.

Graphical user interface, text, application

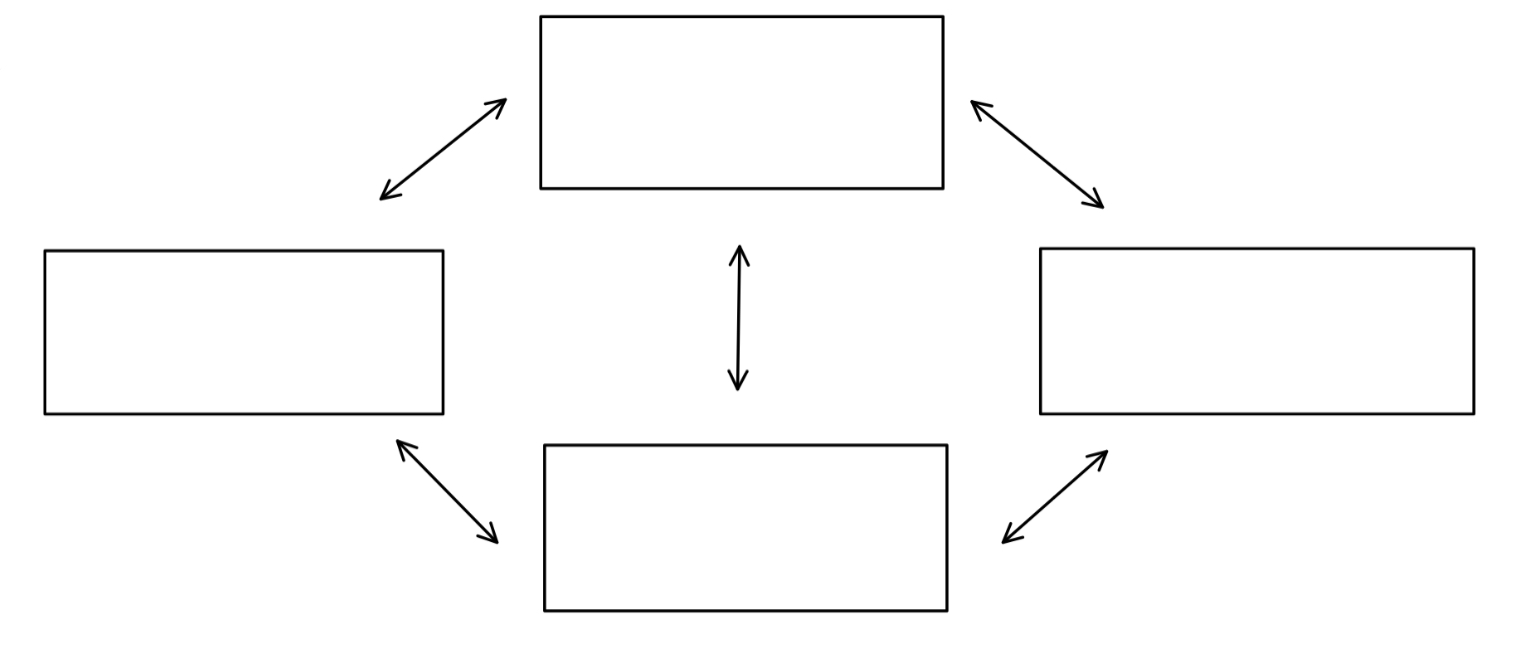
Description automatically generated

*Figure 1. The playfield will have this layout, the green corner will heal while the grey corner turns the robot into a zombie.*

Other guidelines are that the robots themselves must not be larger than 500cm2 in area when playing the game, and no one dimension shall exceed 25cm in length. There also must be at least one limit switch that can detect collision with the walls.

**METHODS:**

In the main code the state that the robot receives information from the radio to specify which state the robots; this in turn changes the has functionality of the robot which is as followed:



**Healthy State**:

Can use Radio to locate balls and other players.

Main objective is to locate & capture the nearest ball.

**Invalid State:**

When both healthy and zombie mode appear, the robot will stop working.

**Healing State:**

Uses radio to find nearest zombie to heal.

**Zombie State:**

Utilizes IR sensor to locate other players.

Can’t communicate with radio

CAPTURES BALL/ HEALS ZOMBIE

GETS TAGGED/HEALED

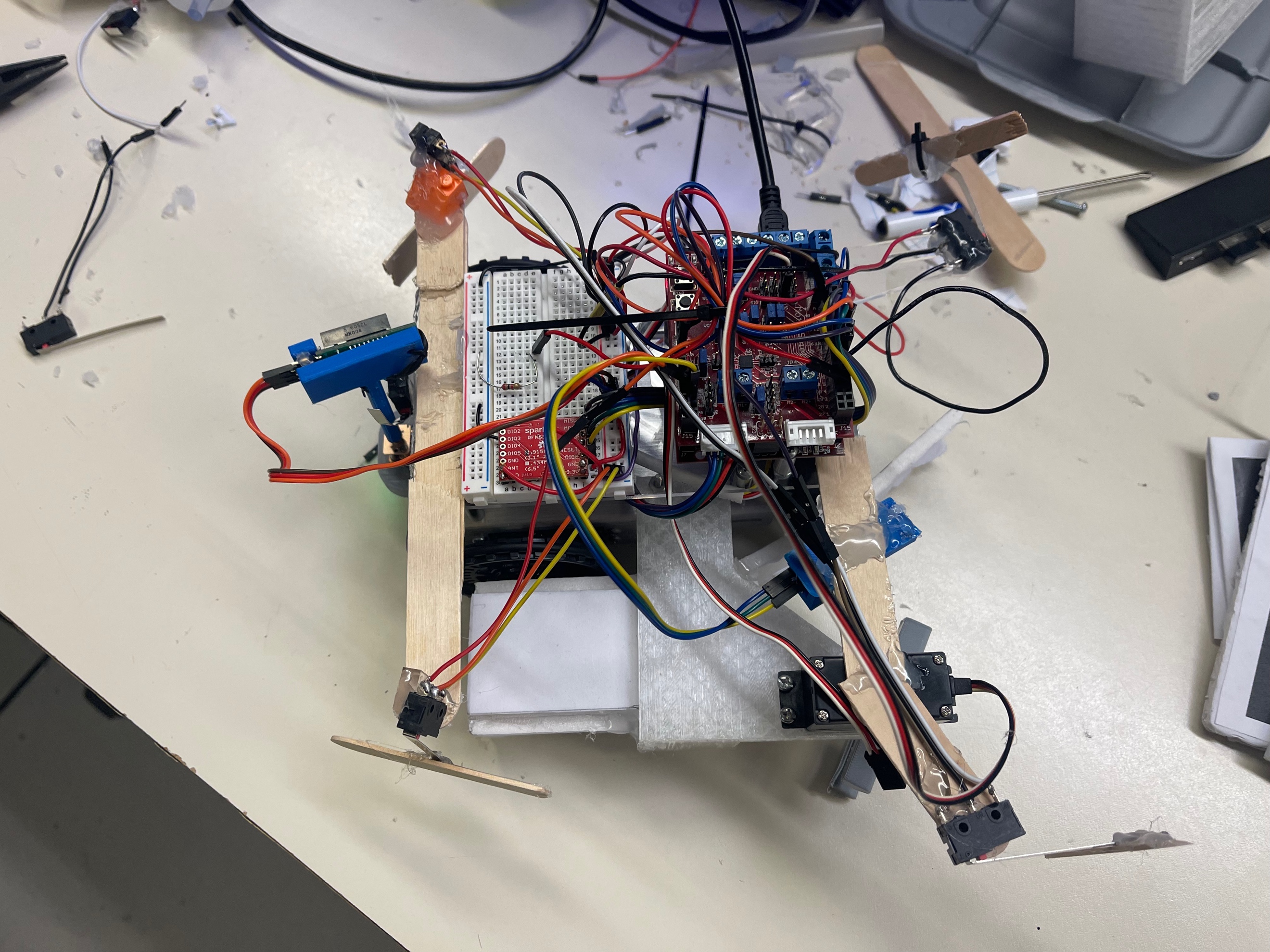
*Figure 2. The invalid behavior occurs when the robot thinks it’s in multiple states which ultimately is an error.*

Diagram

Description automatically generatedFor the movement in general the robot will get the position of a ball or robot via radio transmission.

*Figure 3. The ball will be able to tell whether the target is in front of or behind the robot and will turn until it is facing a robot/ball that is closest to it. From there it will follow the ball/robot with the shortest arch to the target*

The global information will only be used when the robot is in healthy or healing mode. In zombie mode there is an IR sensor that is mounted on a DC motor that will sweep back and forth to detect the nearest robot to it. Once something is detected the robot will know its relative distance from the detected robot via IR sensor and then utilize the DC motors angles to compute the targets local coordinates. From there it will chase the target in arcs exactly like it did in healthy mode. If the distance is deemed too long it will discretize the distance to make the turning easier.



3

2

2

2

1

1

5

1

1

1

4

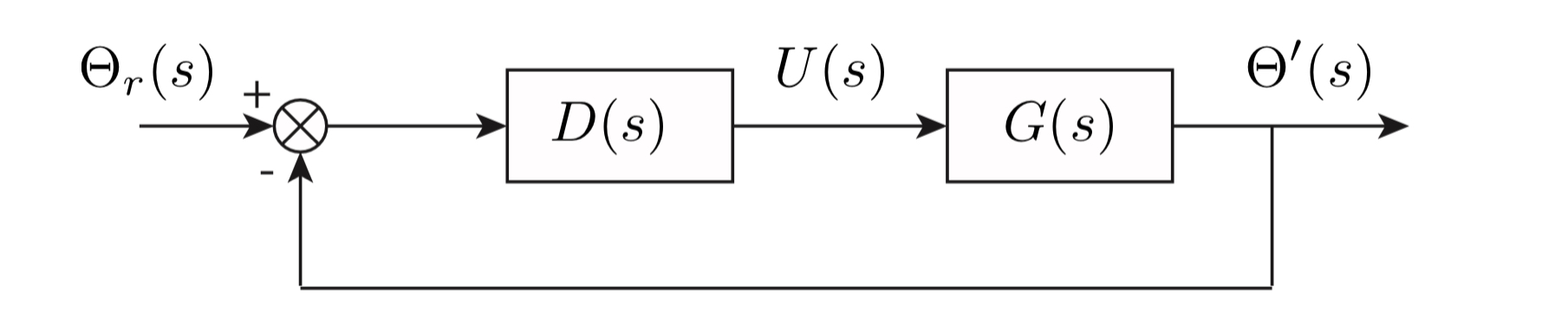
5

*Figure 5. Shows the robots overall layout. The numbers on the picture correspond to the part that is being used for the robot. Note that the mount for the code on top isn’t shown in the picture.*

|  |  |
| --- | --- |
| **PARTS:** |  |
|  | **Limit Switches** |
|  | **Servo Motor** |
|  | **Radio** |
|  | **DC Motor w/ PID controls** |
|  | **IR Sensor** |

The four limit switches that are placed on the four corners of the robot are used for detection. Once the robot hits a wall (presses the switch) the robot will evert away from the wall. Depending on which switch it touches the response will change but the overall objective is to move away from the wall when the switch is pressed.

Since the PID motor has an added mount on the haft the overall equation of motion changes for the motor. Since there is an added mass, the motor must be re-tuned. The input and output relation of the motor is as followed:



*Figure 6. shows the overall loop that the system will follow. D(s) will be changed until a response that is wanted comes out of the motor itself. What is wanted is a desired output angle to match the desired input angle.*

The Ziegler-Nichols method was used to tune the motor since the general G(s) isn’t given. Initially the Proportional gain is raised until the motor response oscillates continuously. When that occurs, the value used for the Kp to make the system oscillate forever is known as the ultimate gain Ku. For this method D(s) is defined with the following equation 1.

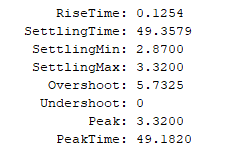
TI & TD are found with the following relationships:

Where Pu is the period of oscillation from the original Ku value. And the control values can be found with the following relationship once the ultimate gain and period are found

The values that were chosen for the controls are in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Control Values | 1104 | 8117 | 37 |

These are the values that we chose to get a response time that we wanted. The overall performance of the response is shown in the following image.



*Figure 7. Shows the response for the motor that’s connected to the IR sensor. We were able to get a response time that’s a fraction of a second long with overshooting under 6%. Overall, the response is exactly what is needed for this application.*

The physical systems response vs time is also graphed to show how quickly the motor will settle to the desired position.

Chart, line chart

Description automatically generated

*Figure 8. Shows the response the engine will have vs time. The overshooting is relatively small for the engine while still being able to settle quickly.*

For the robot itself there was only two parts that were Drawn up and 3d printed. The first attachment would bolt onto the robot and will have a compartment for balls. This was done in two iterations due to the first design being too bulky overall.

A picture containing indoor

Description automatically generated

*Figure 9. Shows the first attachment iteration. This piece was ultimately discarded since adding this will exceed the size restraint for the robot.*

This design was shaved down to fit the size requirements. Since we wanted to be able to make quick fixes to the attachment certain sections of the piece itself were taken out so we could glue on Styrofoam for quick fixes. The finished product of the first mount ended up looking like the following:

A picture containing connector

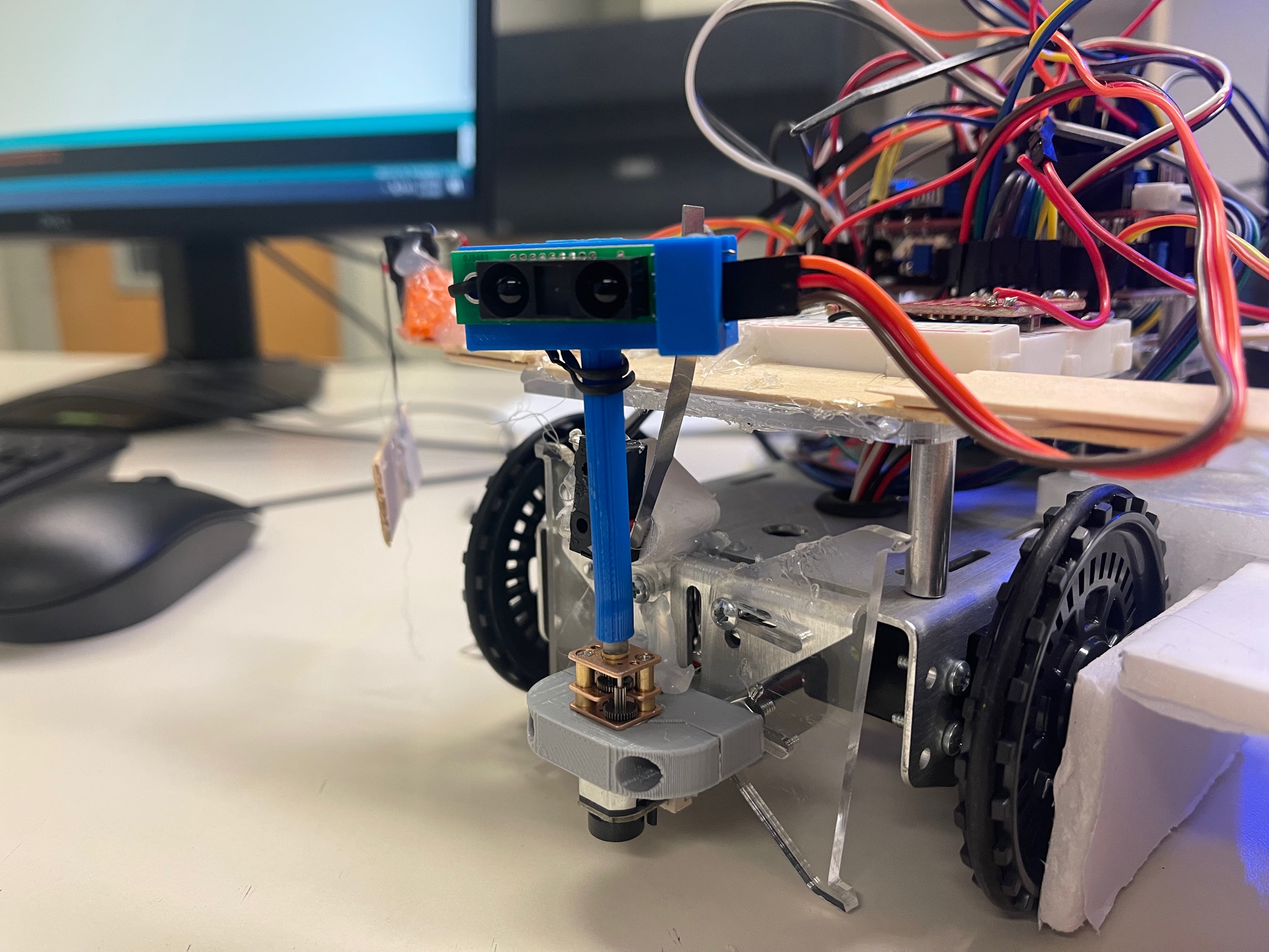
Description automatically generated

*Figure 10. Shows the final iteration of the attachment. Ideally the servo motor will spin a paddle that will help lead the ball into the capture compartment (Styrofoam section).*

Overall, the functionality of the attachment works but if the entire attachment was 3D printed the robot would have more strength. That’s a possible improvement for the future.

The IR sensor that is directly next to the Servo motor with a paddle will detect if there is a ball directly under it. If there is a ball detected servo motors that move the wheels will move in a way to sweep the car over the ball, so the ball is captured more easily. This is necessary because the geometry of the robot might block the overhead camera and the robot can’t collect the ball from that particular point.

The other 3D printed attachment was made to connect the DC motor to the robot, and the DC motor to the IR distance sensor. Prior to this attachment the dc motor was mounted on with bolts and acrylic pieces. but the motor ended up breaking through the plastic or dethatching itself from the IR sensor all together.



*Figure 11. Shows the final clamp that was made for the DC motor as well as the attachment from the DC motor to the IR sensor. The plastic backing is what we originally used to clamp but since it’s so brittle we decided to make it into a backing for the 3D printed motor clamp.*

Overall, these attachments made the sensor shake less which is better for getting sensor data all together. The switch behind the arm is set to zero out the sensor when booting up the robot.

**CONCLUSION:**

The robot was made with the intention to complete all the desired tasks to play Zombie Apocalypse tag. The robot was designed to switch between healthy, healing, and zombie states via radio transmission. Within those states the robot changes functionality, in healthy mode the robot chases a ball, in healing mode the robot chases a zombie, and in zombie mode the robot chases a measured object via IR sensor that is mounted on a DC motor that sweeps the playfield. In healthy mode the robot is programed to stay away from other robots and zombies. Specific collision switches were placed at each corner of the robot to enable the robot to get out of corners or walls when it runs into them. The size requirements were taken into consideration and met for the most part. All in all, a robot was designed to complete the tasks given by the game, and was able to execute every task specified.