# ECE 445

Senior Design Laboratory Design Document

# Autonomous Golf Green Divot Locator Robot

**Project Team #19**

Akhil Bonela (abonela2), Michael Cherry (mcherry3), Ved Eti (vedeti2)

# Contents

# 1. Introduction

## 1.1 Problem

Preserving the quality of golf greens is essential to ensuring a fair and enjoyable golfing experience. However, a common challenge that undermines this objective is the failure to repair ball marks. When a golfer's ball lands on the green, it creates a small indentation, or divot, in the surface. While it is customary for players to use a repair tool to fix these marks, not all golfers adhere to this etiquette. This neglect leads to an increase in divots and uneven patches on the green, which can have detrimental consequences for both the course and the golfers.

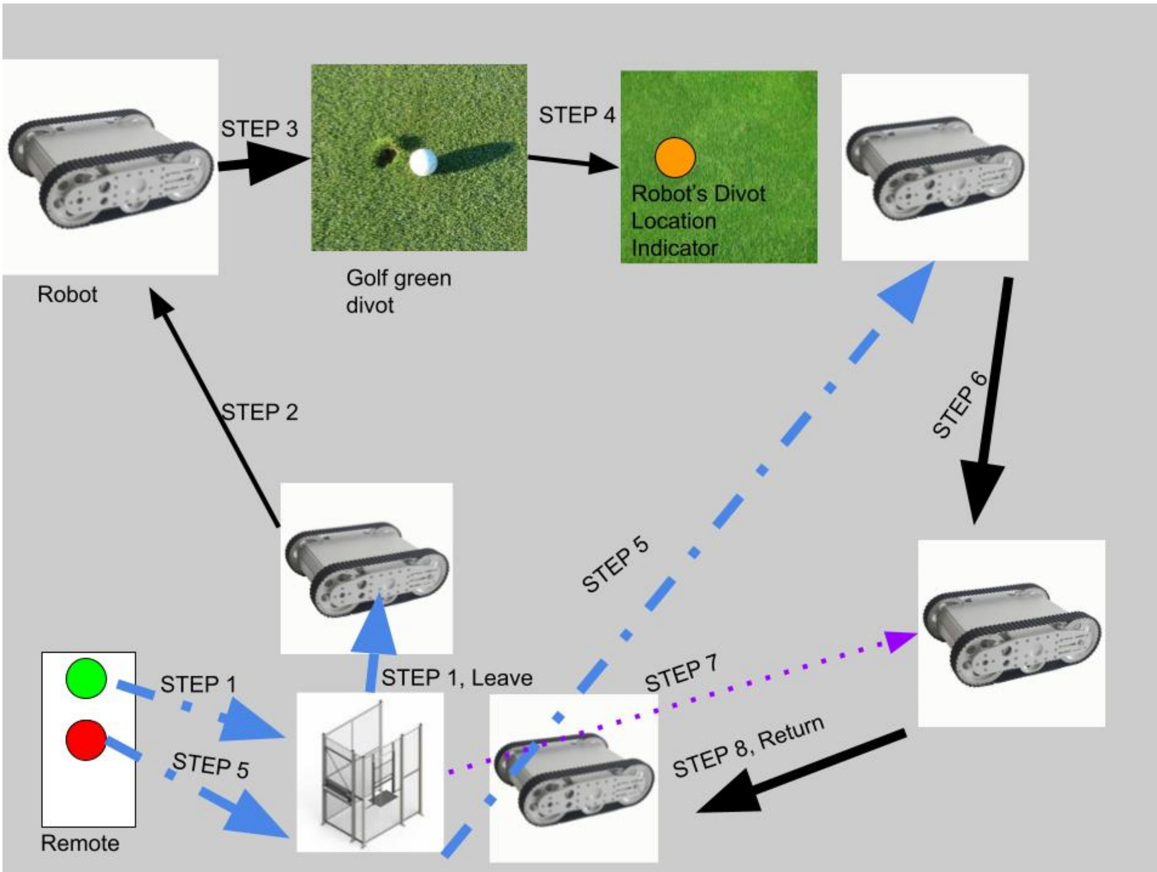
Unrepaired ball marks can significantly diminish the quality of a golf green. The divots disrupt the smooth flow of putts, making it more difficult for golfers to judge distances and control their shots accurately. Additionally, these marks can interfere with the green's drainage system, leading to localized water pooling and potential turf damage. Furthermore, the unsightly appearance of a green littered with divots can diminish the overall aesthetic appeal of the course, negatively impacting the golfing experience for all players.

## 1.2 Solution

Our proposed solution involves developing an autonomous robot equipped with advanced sensing and marking capabilities. This robot will be designed to traverse the golf green at the end of the day, when the golf club closes. Using stereo cameras, the robot will accurately locate divots by analyzing the differences in depth between the surrounding turf and the indented areas.

Once divots are identified, the robot will use a custom-designed marking tool to clearly indicate their locations. This tool will leave a visible mark on the green, guiding golfers to repair the divots before their next shot. By automating the process of divot identification and marking, our robot will significantly reduce the manual effort required to maintain the quality of golf greens while ensuring that all divots are promptly addressed

## 1.3 Visual Aid




## 1.4 High Level Requirements

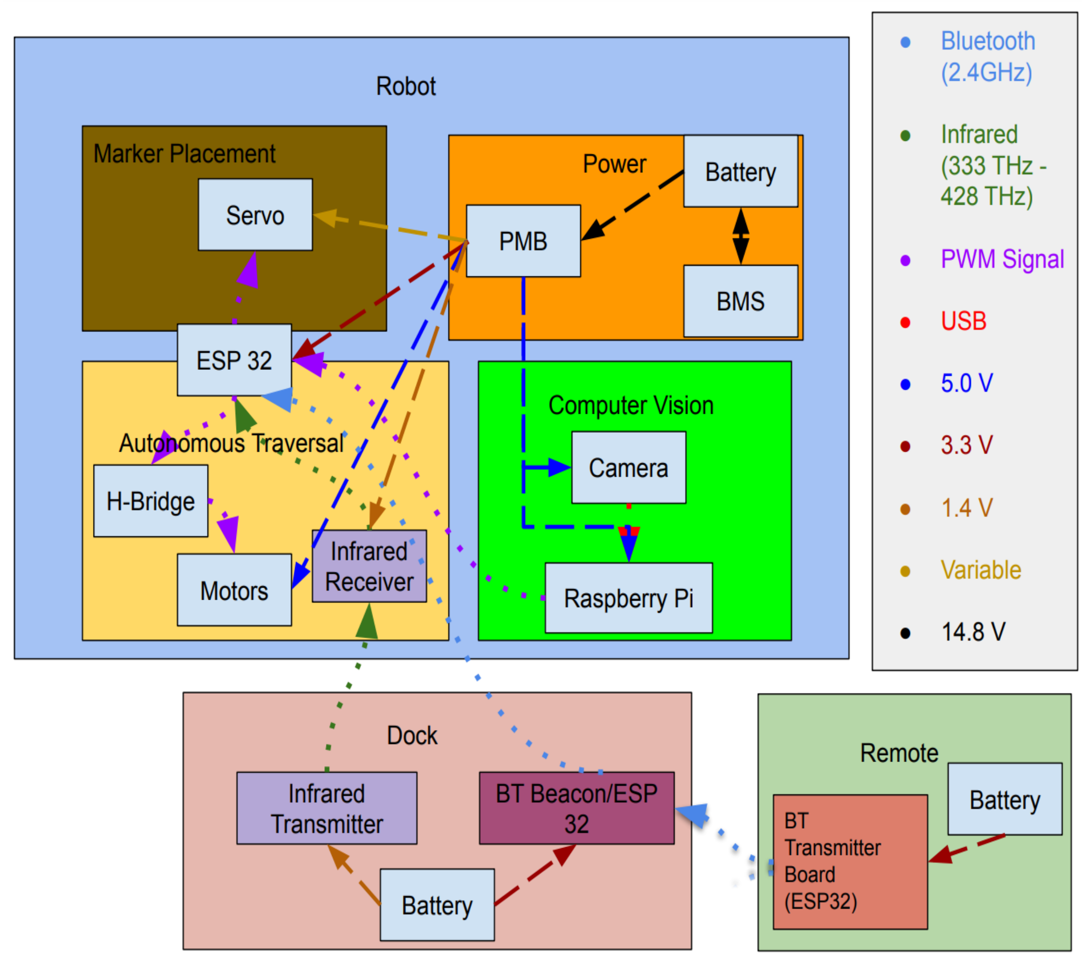
1. Be able to create a robot that is compact and light enough to traverse the golf course green without damaging the golf green
   a. Want size to be with in 18”x18”x18” and have it weigh under 15lbs

2. Have the robot be able to effectively indicate to the user, either visually or programmatically, where it “thinks” a hole/divot in the golf green is, and place a marker, accurate to within +/- 3 inches
   a. The diameter and depth we hope to successfully locate and indicate is about 1-1.5 inches in radius and 0.5-0.75 inches in depth

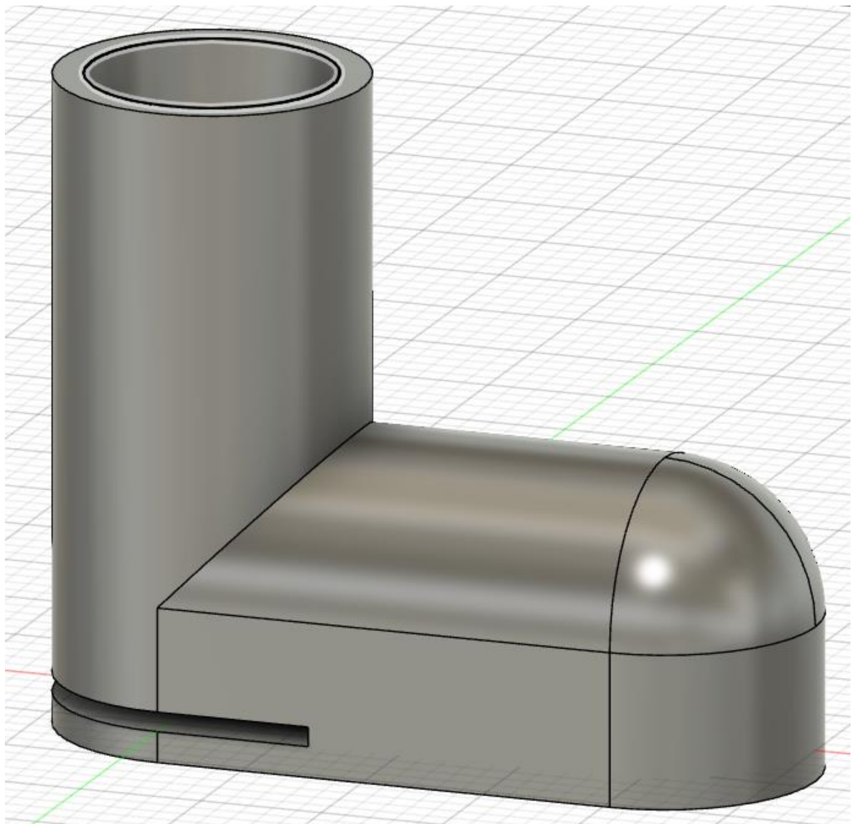
3. Have the dock/cage successfully transmit signal to the robot to return on press of user's remote, and be accurate to within +/- 6 inches of the center of the cage/dock

# 2. Design

## 2.1 Block Diagram:



2.2 Physical Design (if applicable):



*Draft Model of Vertical Marker Dispenser*

*Draft Model of robot chassis*

Our physical design composes of the robot chassis, its treads, and the marker dispenser. We also have two smaller physical designs for the dock and the remote. These components will be critical to achieving the traversal and dispensing accuracy requirements.

The placement of infrared sensors and transmitters on the dock and robot may need to be placed higher up or more in front of their respective builds, to ensure a clear path for infrared sensing is present.

## 2.3 Subsystem Functional Overview & Requirements

### 2.3.1 Control Subsystem

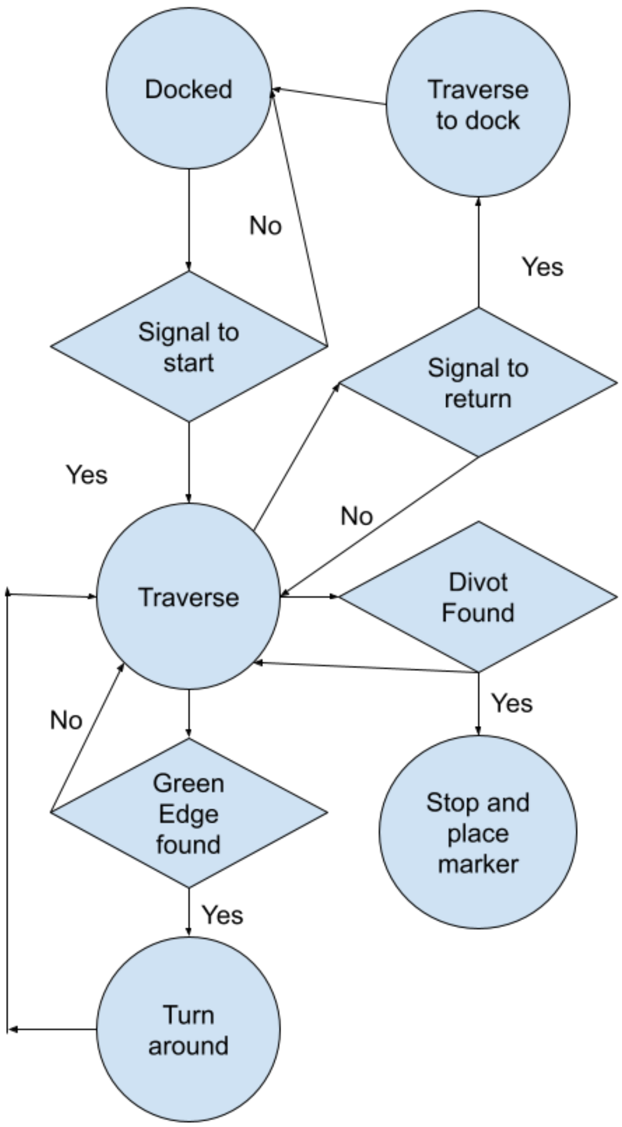
This subsystem controls the robot's movement as it traverses the golf green. This subsystem is mostly going to be interfacing with our microcontroller (ESP 32), our motors, an H-Bridge chip, and the Raspberry Pi. All of the computer vision tasks will be performed on the Raspberry Pi, as our microcontroller will not have enough computational power. The output of the computer vision to either detect the edges of the golf green or a divot will then be sent to the ESP32 microcontroller through general GPIO pins. The microcontroller will then give out instructions for the robot if it detects a golf green, interacting with an H-Bridge motor driver, allowing us to turn around and change the direction of the robot, and continue sweeping the area. This will act very similar to a common Roomba, and other robotic vacuum cleaners. We repeat this process until we traverse and check the entire green.

We plan on using a pre-produced chassis listed above in our component list so that we don't have to spend time making and manufacturing our own chassis with motors. We will add our own microcontroller, PCB for power distribution, and battery to the chassis, and use the chassis mostly for the mechanical build.

The main bridge between this subsystem and the rest of the subsystems is the ESP32 microcontroller and the Power distribution board. The ESP32 microcontroller will receive signals from the Raspberry Pi that detail signals of finding either a divot or detecting the edge of the golf green. We plan to use two general purpose input output (GPIO) pins from the Pi and connect it to the microcontroller. These signals will be a simple binary signal (On/Off) to specify if a divot is found or if the edge of the grass is detected. The microcontroller will be programmed to send respective signals to the H-bridge motor controller; either stop if a divot is detected or turn the robot around if the edge of the green is detected. Please see the computer vision subsystem module description for more information on how the raspberry pi and sensors will be used as detection devices. This subsystem is vital to completing the high-level requirements, including the return to the dock capability, traversing the green without causing damage, and stopping to place a marker at the location of the divot. Removing any of the components (Microcontroller, motors, H-Bridge) will cause failure of the system, and will not meet our high-level requirements. Removal of motors will cause our robot to be stationary, not allowing us to traverse the green and mark the divots. Removal of the H-bridge motor driver will not allow us to control the motors effectively using the microcontroller and will require using either a Raspberry Pi or multiple microcontrollers. The microcontroller is extremely important, as it is the component that takes the detection signals and tells the motors and h-bridge how to maneuver the robot. The Raspberry Pi can also be used in place of the microcontroller and would have been our preferred choice but was not allowed for this project.

The last main part of the autonomous traversal is an infrared receiver. This fulfills the last high-level requirement, being able to return to the dock with a level of accuracy. This part is connected to the microcontroller and will direct the robot to move towards the dock. More information on how exactly this will work is given in the dock / remote control subsystem module. Please see power subsystem requirements for details on specific power requirements of components in the autonomous traversal and control subsystem.

| | |
|---|---|
| • Limit a top speed of 0.61 meters per second to keep in line with tolerance analysis calculation. | • Set a track distance of 10 meters.<br>• Use a timer to see how long the robot takes to travel 10 meters.<br>• Calculate and ensure the speed is less than the specified amount. |
| • Stop within 3 inches of where a signal to stop is sent to the microcontroller. | • Program microcontroller to send a stop signal after 10 seconds of top speed.<br>• Keep track of the 10 seconds an a stopwatch and location of robot at that time.<br>• Make sure this is within 3 inches of final stopping position. |
| • Be able to complete a 180-degree turn when asked to do so. | • Send binary-on signal to respective microcontroller pin.<br>• Ensure the robot turns around, but also is in a different lane to its current position. |




Above is a flowchart representation of our entire control and traversal subsystem getting inputs and encountering various scenarios.

## 2.3.2 Computer Vision Subsystem

The image processing module will mostly have two tasks, identify divots, and identify edges of golf greens. The two components of the computer vision/image processing submodule are the Raspberry Pi and a stereo camera. The Raspberry Pi will get live image feed from our camera and will perform some computer vision algorithms for detection and classification. It will pass along information about what it detects to the ESP 32 microcontroller so that we can either use the traversal module to move the robot, or the marker placement module to place markers down. For the divot sensing, we plan on using the stereo camera as a method to estimate the changes in depth. As per our research, we plan on using semi-global block matching for our depth perception, which seems like can tell the difference between uneven surfaces on the ground. As another option, we will also take a look at using a yolo v8 model in order to increase the speed at which we create our inferences. We will also use the camera feed and algorithms for the green detection to make sure the robot does not go outside the green. For this we plan on using some thresholding and segmentation to differentiate between the current surface it is on and the other surface it sees. We plan on using various Python libraries such as PyTorch and OpenCV, which are both too computationally intensive for a microcontroller.

As mentioned before, the two components of image processing are the stereo camera and a Raspberry Pi. The stereo camera that we plan on using will be connected to the Raspberry Pi through a USB port. Depending on whether the algorithm detects a divot or if the robot is veering off of the green the Pi will have two GPIO pins that will connect directly to the microcontroller via a simple binary on/off signal. Please see power subsystem for extended details on the power requirements of computer vision subsystem.

| | |
|---|---|
| • Be able to identify divots with 90 percent accuracy. | • Gather pictures of divots and regular grass online and train model.<br>• Retain some pictures to use as test dataset.<br>• Ensure that 90 percent of all divot pictures are marked as a divot.<br>• Ensure that 90 percent of all grass pictures are marked as grass. |
| • Create divot detection algorithm with a run time of less than 75 ms per inference. | • Use python time module to start timer when new image is read from file system or camera feed.<br>• Use time module to end timer when a final prediction is made on the image.<br>Send binary (On/Off) signal to GPIO pins if divot is detected. Use probe to ensure pins are correct values. |

| | |
|---|---|
| • Create green edge detection algorithm with a run time of less than 150 ms per inference. | • Use python time module to start timer when new image is read from file system or camera feed.<br>• Use time module to end timer when a final prediction is made on the image.<br>• Send binary (On/Off) signal to GPIO pins if green edge is detected. Use probe to ensure pins are correct values. |

### 2.3.3 Power Subsystem

The power subsystem is a critical component of the autonomous divot repair robot, responsible for providing reliable and consistent electrical power to all its components. The robot portion of the system comprises a Lithium Polymer (LiPo) battery pack, a Battery Management System (BMS), and a Power Management Board (PMB). These components work together to ensure efficient and safe power distribution.

The LiPo battery, a Zeee 14.8V 4S Lipo Battery (50C 3300mAh), offers several advantages. Its high energy density allows for a compact and lightweight design, making it suitable for mobile applications. The 50C discharge rating ensures the battery can deliver sufficient power for demanding tasks.

The BMS safeguards the battery pack from various issues, including overcharging, over-discharging, short circuits, and overcurrent. This protection is crucial for maintaining battery longevity and preventing safety hazards.

The PMB, a custom-designed PCB, plays a vital role in distributing regulated power to all subsystems. It employs buck converters for 5V and linear regulators for 3.3V, ensuring efficient power conversion and voltage regulation. Additionally, current limiting circuitry is integrated into the PMB to protect components from excessive current draw.

The dock and remote utilize similar power sources: 3.7V 3000mAh Rechargeable Lithium-ion (Li-ion) batteries. These batteries provide reliable and long-lasting power for their respective components. Voltage regulators within the dock and remote ensure stable 3.3V supplies to the ESP32 microcontrollers.

| **Requirement** | **Verification** |
|---|---|

| | |
|---|---|
| • Power Subsystem in robot must provide a regulated 5V output to the Raspberry Pi, Stereo Camera, and Ultrasonic Sensor | • Use a digital multimeter (DMM) to measure the output voltage of the PMB connected to these components.<br>• Ensure the voltage remains within the specified tolerance (e.g., 4.9V - 5.1V) under varying load conditions (e.g., idle, peak usage).<br>• Conduct measurements with different current loads to assess the PMB's regulation capabilities. |
| • Power Subsystem in robot must provide a regulated 3.3V output to the Microcontroller | • Use a DMM to measure the output voltage of the PMB connected to the microcontroller.<br>• Ensure the voltage remains within the specified tolerance (e.g., 3.2V - 3.4V) under varying load conditions.<br>• Conduct measurements with different current loads to assess the PMB's regulation capabilities. |
| • Power Subsystem in robot must be able to supply a minimum of 2352 mA | • Use a DMM to measure the current drawn by the system under peak load conditions (e.g., when all components are operating at their maximum power).<br>• Ensure the current output exceeds the minimum requirement of 2352 mA. |
| • Battery must maintain a voltage within the specified range (11.1V - 14.8V) | • Use a DMM to monitor the battery voltage during operation.<br>• Ensure the voltage remains within the specified range under varying load conditions.<br>• Conduct regular battery health checks to assess its capacity and performance. |
| • Must operate for at least 2 hours on a fully charged battery | • Conduct a runtime test under typical operating conditions, simulating various tasks and power demands.<br>• Record the battery voltage and current draw throughout the test.<br>• Calculate the actual runtime and compare it to the target of 2 hours. |
| • Dock and Remote must operate for at least 4 hours on a fully charged battery | • Conduct runtime tests for the dock and remote under typical operating conditions.<br>• Record the battery voltage and current draw throughout the tests. |

| | |
|---|---|
| | • Calculate the actual runtime for each device and compare it to the target of 4 hours. |

Component Power Requirements:

| **Components** | ***Typical Operating Voltage (V)*** | ***Typical Operating Current (mA)*** | ***Required Voltage (V)*** |
|---|---|---|---|
| Stereo Camera | 5.0 | 250 | 5.0 |
| Micro-Controller (ESP32-S3) | 3.3 | 80 (active)/20 (idle) | 3.3 |
| Raspberry Pi (Model 4) | 5.0 | 3000 (peak)/500 (idle) | 5.0 |
| Servo Motor | Variable | 500 (peak)/100 (holding) | Variable |
| TT DC Motor | 5.0 | 250 | 5.0 |
| Infrared Emitter and Receiver Diode | 1.4 | 30 | 1.4 |

## 2.3.4 Marker Placement Subsystem

The purpose of the marker placement subsystem is to facilitate the communication between our vertical marker dispenser and the robot's esp32 microcontroller. This is the subsystem that will make it possible for our robot to indicate where it detected a divot and leave a visible marker on the golf green we can use to directly compare its predictions with actual divots. When the computer vision subsystem detects a divot using its segmentation algorithm, the ESP32 will receive a HIGH signal from the Raspberry Pi once the robot is moved into position. Upon receiving the Pi's signal, the ESP32 will send a serial Pulse Width Modulation (PWM) signal from one of its General Purpose Input/Output (GPIO) pins to indicate to the robot that it is time to dispense a marker from the vertical dispenser. The servo will be rotating a small fin that will move out from under the vertical dispenser containing our markers. This will make a marker fall down, and upon returning to under the vertical dispenser, it will push the marker down a second vertical tube to drop the marker in a static position relative to the body of the robot and ensure a flat landing of the marker.

| **Requirement** | **Verification** |
|---|---|

| | |
|---|---|
| • When the robot detects a divot and moves into position, the robot dispenses a location indicator marker within 10 seconds | • Ensure the robot is in a state where it detects a divot and moves into position relative to the vertical dispenser being on top of the location it detected. Or, hardcode the above conditions to be true, so that the programmatic logic for the dispensing to start is met.<br>• Once this condition is met, we can measure the voltage of the ESP32 GPIO pin connected to the servo to determine if it is transmitting a PWM signal to the servo<br>• Once this condition is met, we can see if the servo actually rotates and is able to successfully dispense a marker, timed within a matter of 10 seconds |
| • When the servo dispenses a marker, the marker lands flat and within an inch of directly under the dispenser | • Ensure the servo is receiving a PWM signal from the ESP32 GPIO pin. This can be hard coded or achieved organically by detecting a divot and having the robot line up the dispenser.<br>• Keep the vertical dispenser stationary and allow the servo to rotate to dispense a marker<br>• Observe the dispensing to ensure that the marker falls out and lands flatly.<br>• Measure the (x,y) location of the marker and compare it to that of the vertical dispenser. (This measurement/comparison can be done in a few ways) |

### 2.3.5 Remote Control & Dock Subsystem

The role of the remote control and dock subsystem is to allow users to remotely communicate with the robot and to give users the power to send the robot out onto the green or back to its home location on their command. This is the driving subsystem for the robot to be able to accurately return to the center of the cage, within an error of 6 inches. The remote will contain its own ESP32 that will function as a Bluetooth client, transmitting 2.4GHz signals to the dock to indicate the user wants the robot to begin traversal. The dock itself will also contain its own ESP32 that will function as both a Bluetooth transmitter and receiver. The dock's ESP32 will receive a Bluetooth signal from the remote and proceed to turn send a Bluetooth signal to the robot to instruct it to run the code to begin traversing the green and detecting divots. We made the dock have its own ESP32 subsystem because it will also include an infrared transmitter to interface with the robot. We have used both Bluetooth and infrared signals for the dock and robot communication to make use of the reliable distance of Bluetooth and the accuracy of infrared to best meet our high-level requirement for dock returning accuracy.

| **Requirement** | **Verification** |
| --- | --- |
| • The remote can successfully command the robot to begin traversal within 10 seconds of user press | • Ensure the robot is in its dock and is in an idle waiting state.<br>• Press the Start button on the remote and begin a timer.<br>• Measure for a Bluetooth signal received on the ESP32 of the dock<br>• Check the ESP32 is outputting the Bluetooth signal to indicate the robot to move.<br>• Measure on the ESP32 of the robot that the Bluetooth signal was properly received from the dock.<br>• Observe the robot to determine if it successfully begins moving, within ten seconds of the timer starting |
| • The remote can successfully command the robot to begin returning to dock within 30 seconds | • Ensure the robot outside its dock and is not in an idle waiting state.<br>• Press the Return button on the remote and begin a timer.<br>• Measure the ESP32 of the remote to determine if it properly sends the return signal to the dock.<br>• Measure the ESP32 of the dock to see if it received the remote's Bluetooth signal, and measure if the signal to the robot is transmitted after.<br>• Measure the ESP32 of the dock begins IR transmission when the robot is closer to the dock (IF APPLICABLE)<br>• Measure the ESP32 of the robot to see if it is receiving the Bluetooth and IR signals of the dock.<br>• Observe the robot begins making progress towards the dock, within a matter of 30 seconds |
| • When the user wants the robot wants to stop traversing, the press of the emergency stop | • Ensure the robot is in a non-idle state |

| | |
|---|---|
| button on the remote makes the robot stationary, within 5 seconds | • Press the STOP button on the remote and begin a timer.<br>• Measure for a Bluetooth signal received on the ESP32 of the dock<br>• Check the ESP32 is outputting the Bluetooth signal to indicate the robot to stop.<br>• Measure on the ESP32 of the robot that the Bluetooth signal was properly received from the dock.<br>• Observe the robot to determine if it successfully stops moving, within five seconds of the timer starting |

## 2.4 Tolerance Analysis:

Through discussions with our TA Pusong, we identified that the main subsystem that will affect the success of the project is the vision subsystem, specifically with the speed, complexity, and accuracy of the computer vision algorithms. The efficiency of the algorithm to process camera feed and classify divots will then affect how close a marker is placed to the actual divot. While it is hard to give a complete estimate on the speed of our algorithm on the Raspberry Pi, through prior coursework and projects working with the Pi, a reasonable inference time will be about 100 ms. Using a 25 percent buffer, we can assume that the maximum amount of time that will be used is 125 ms, this includes any other time delay to get camera feed to the vision algorithm, and raising signal flags to the microcontroller. Using the example inference time of 125 ms, we can now calculate the speed at which the robot should travel so that we can meet our high-level requirement of placing the marker within 3 inches of the divot. The speed formula that we use is given as following:

$$S = D / T$$

$$S = 3 \text{ inches} / 125 \text{ ms} = 0.024 \text{ inches/ms}$$

The speed at which we can travel to be reasonably safe with our marker placement is 3/125 inches/ms otherwise 0.024 inches per millisecond.

$$S = 0.024 \text{ inches/ms} * 25.4 = 0.61 \text{ m/sec}$$

Converting this to metric units will give us 0.61 m/sec as a top speed at which our robot can go to account for any delay in our processing pipeline. We might opt to go slower than this speed for any safety concerns, but will not go over this speed at all.

In addition, we have also provided a tolerance analysis for our power subsystem to ensure all components are sufficiently powered to work, including robot, docking station, and remote control. If any of these systems fail, and do not have enough power / or too much power to continue working as intended,o our project will not be able to work as intended. To combat this, we plan on using a printed circuit board that acts as a power distribution board to regulate the voltage. Since we have components that operate in different voltages (3.3 V, 5 V,6 V), we need to add in some voltage regulators. For example to calculate the power dissipation of such a regulator can be found by using the following formula:

$$Pd = Iout*(Vin - Vout).$$

In addition to the voltage, it is also necessary to regulate the amount of current supplied to each component. As calculated, the minimum current required to power all of the components is 250 + 80 + 500 + 100 + 4*250 + 30. We also want to make sure that we have some sort of buffer to make sure that the current does not fall below the required current to ensure this we would have enough current to ensure smooth operation. We calculated keeping a 20 percent buffer would be sufficient to make sure all components are properly powered, giving us at least 2352 mA should be supplied to the system at all times.

# 3. Cost & Schedule

## 3.1 Cost Analysis

### 3.1.1 Labor Costs

A reasonable hourly pay rate for an ECE graduate is about $37/hr. We plan on spending at least 10 hours every week on the project. With three people, our total labor costs will be

*Total Labor Costs = 3 members * 10 hours/week/member * 8 weeks * $37/hour = $8,880*

We must also multiply the labor cost by a factor of 2.5 to account for any overhead.

*$8,880 * 2.5 = $22,200*

### 3.1.2 Part Costs

| Part | Quantity | Cost | Link |
|---|---|---|---|
| **TSOP38238 IR Receiver & TSAL6200 IR Transmitter** | 1 | $8.49 | Link |
| **SG90 Servo Motor** | 1 | $8.99 | Link |
| **ESP32-S3-WROOM-1 Microcontroller Chip** | 2 | $3.20 | Link |
| **Raspberry Pi 4** | 1 | $55 | Link |
| **Stereo Camera** | 1 | $46.00 | Link |
| **Robot Chassis** | 1 | $22.99 | Link |
| **Tactile Push Button Switch 6x6x 6mm** | 1 | $5.69 | Link |
| **14.8V 4S Lipo Battery** | 1 | $33.99 | Link |
| **3.7 Volt Rechargeable Battery** | 1 | $11.98 | Link |
| **Lithium Charger for 3.7V Rechargeable Batteries** | 1 | $7.99 | Link |
| **BMS Protection Board** | 1 | $8.99 | Link |
| **LiPo battery charger** | 1 | $16.99 | Link |

### 3.1.3 Total Costs

Our raw parts cost $264.39. Including a 5% shipping cost and 10% tax rate, we estimate the final cost of all parts is $288.39. Adding our labor cost of $30,000 to our final parts cost, we get a final cost of $30,288.39.

## 3.2 Schedule

| Week | Task | Person |
|---|---|---|
| ***October 6 – October 12*** | 1. Design PMB PCB<br>2. Research Computer Vision Algorithms and optimization<br>3. Design Dock PCB & Remote PCB | 1. Akhil Bonela<br>2. Ved Eti<br>3. Michael Cherry, Akhil Bonela |
| ***October 13 – October 19*** | 1. Teamwork Evaluation<br>2. Design Marker Placement Device<br>3. Implement segmentation algorithm<br>4. Develop robot physical design<br>5. Buy Parts | 1. Everyone<br>2. Akhil Bonela, Michael Cherry<br>3. Ved Eti<br>4. Everyone<br>5. Everyone |
| ***October 20 – October 26*** | 1. Assemble robot physical design<br>2. Solder & Test PCBs<br>3. Update Dock PCB & Remote PCB & PMB<br>4. Start programming microcontroller<br>5. Implement divot classification | 1. Everyone<br>2. Michael Cherry, Akhil Bonela<br>3. Michael Cherry, Akhil Bonela<br>4. Ved Eti, Micheal Cherry<br>5. Ved Eti |
| ***October 27 – November 2*** | 1. Test, debug, update robot's physical design<br>2. Computer Vision debug and optimization<br>3. Microcontroller programming | 1. Everyone<br>2. Ved Eti<br>3. Ved Eti, Micheal Cherry, Akhil Bonela |
| ***November 3 – November 9*** | 1. Refine robot's physical design<br>2. Microcontroller programming | 1. Everyone<br>2. Everyone |
| ***November 10 – November 16*** | 1. Debug and Optimize Computer vision models<br>2. Debug motor firmware | 1. Ved Eti<br>2. Akhil Bonela, Micheal Cherry |
| ***November 17 – November 23*** | 1. Mock Demo<br>2. Final debugging | 1. Everyone<br>2. Everyone |
| ***November 24 – November 30*** | Fall Break | Fall Break |

| | | |
|---|---|---|
| | 1. Last Minute Debugging<br>2. Prepare for Final Demo, Presentation, and Paper | 1. Everyone<br>2. Everyone |
| ***December 1 – December 7*** | 1. Final Demo | 1. Everyone |
| ***December 8 – December 14*** | 1. Final Presentation and paper | 1. Everyone |

# 4. Discussion of Ethics & Safety

The development of an autonomous golf course robot raises important ethical and safety concerns. To address these concerns, we must carefully consider the potential impacts of the robot's operation and implement appropriate safeguards.

From an ethical standpoint, the robot's actions must be accountable and transparent. We will establish clear guidelines for the robot's operation and ensure that it is used responsibly. Additionally, we will address environmental concerns by selecting eco-friendly components and minimizing the robot's impact on the golf course.

Safety is another critical factor. The robot must be designed and operated to prevent accidents and injuries. We will adhere to relevant safety standards, implement obstacle avoidance systems, and provide emergency stop mechanisms. Furthermore, we will ensure that the robot does not pose a risk to human safety or property damage.

To mitigate potential risks, we will implement several solutions and safeguards. These include limiting the robot's size and weight to prevent damage to the golf course, using rubber edges to protect against collisions, restricting the robot's speed, and providing a user-controlled emergency stop feature. Additionally, we will incorporate a battery management system (BMS) to monitor the health of the LiPo battery and prevent potential hazards.

Furthermore, our team will adhere to the IEEE's Code of Ethics throughout the project. This includes principles such as acting in the best interest of the public, avoiding harm to others, maintaining honesty and integrity, and respecting the environment. We will ensure that our design and development processes align with these ethical principles.

By following the IEEE's Code of Ethics, our team will strive to create a responsible and beneficial autonomous golf course robot. We will prioritize safety, environmental sustainability, and ethical considerations in all aspects of our work.

# 5. Citations


[1] Ultralytics. "Yolov8." *YOLOv8 - Ultralytics YOLO Docs*, 30 Sept. 2024, docs.ultralytics.com/models/yolov8/#supported-tasks-and-modes.

[2] Espressif Systems, "ESP32-S3-WROOM-1 and ESP32-S3-WROOM-1U datasheet," accessed: Oct. 1, 2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[3] Tower Pro, "SG90 Micro Servo Motor datasheet," accessed: Oct. 1, 2024. [Online]. Available: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

[4] IEEE, "IEEE Code of Ethics," accessed: Oct. 3, 2024. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html