WORCESTER POLYTECHNIC INSTITUTE

ROBOTICS ENGINEERING PROGRAM

Limited Operational System for The Autonomous Removal of Candlelight

RBE 2002 Final Project Report

SUBMITTED BY

Nick Lanotte

Parker Grant

Dillon Arnold

Benjamin Wagner

Date Submitted	**: May 3, 2017**
Date Completed	**: April 29, 2017**
Course Instructor	**: Prof. Putnam**
Lab Section	**: C0x**

Abstract

In 2015, 365,500 home fires were reported, according to the National Fire Protection Agency. Efforts must be made to develop a system, or series of systems, to autonomously monitor and deal with any incidental fires that occur in homes across the United States. In an effort to simulate a solution, students in RBE 2002 develop robots within certain restrictions in order to autonomously navigate an unknown field, locate an open flame in the form of a candle and extinguish said flame. The LOSt ARC couples continuous scanning with wall-following navigation to locate said flame with a Wii remote and extinguish it with a relatively large, high-volume fan.

Contents

Section 1: Introduction

According to the National Fire Protection Agency, there was an estimated 365,500 fires reported in 2015 [4]. While this is less than the reported 381,000 fires of 2005 [4], a significant number of deaths and injuries occur due to cooking and heating equipment left unattended [3]. To combat this, some system capable of autonomous monitoring and extinguishing of small fires must be implemented into the common American household. Such an autonomous system would function in a similar way to the iRobot Roomba- with regards to its independence, navigation, and overall functionality- except it would focus on extinguishing or delaying the spread of small fires rather than the cleanliness of a room.

In an effort to simulate this task, the RBE 2002 term project is designed around the idea of a robot fighting a fire in an unknown environment. As is the case with real incidents, the fire may not always be immediately visible from the starting location of the robot. Therefore, a firefighting robot must be able to autonomously locate, navigate to and extinguish the fire with minimal human interaction. If at all possible, the robot should be able to return to its original starting position, similar to the most current model of Roomba vacuums. Additionally, it is necessary for such a robot to announce in some way that the fire has been located, and subsequently extinguished, as well as the location of the fire in relation to the robot's starting point, such that the homeowners could know when they must leave, if it is safe to enter, and where the fire originated. [5]

In the case of the aforementioned term project, the majority of the requirements for a proper firefighting robot experience a reduction in some form. With the term project, the fire to be extinguished is a candle between 10 and 30 centimeters, instead of an outlet or appliance on fire. In lieu of an entire household, a square field eight feet to a side was used to test the robot's ability to accomplish the task at hand. Considering that most appropriate forms of household fire extinguishers utilize a condensed form of fire-fighting gas, such as CO_2, or a foam of some sort, which can either be deadly or messy in large amounts [1], and the field would be used by multiple teams, robot designs were encouraged to find other solutions to extinguishing the flame. Additionally, teams were not given an abundant amount of time to complete the task. Since a full scale fire can become life-threatening in as little as two minutes and engulf an entire residence in as little as five [2], teams were allotted only five total minutes to locate and extinguish the flame. [5]

Section 2: Methodology

From the details laid out in the introduction above, several key aspects of an appropriate robot would need to be implemented. First and foremost, a method of navigating reliably and safely around the field was needed for the robot. Next, a solution to extinguishing the flame needed to be finalized. Following these two requirements, the necessary sensors and subsequent state programming were developed in order to complete the robot. With these four main components finalized, the robot was able to complete the task at hand. A computer generated model of the robot can be seen in figure 1.

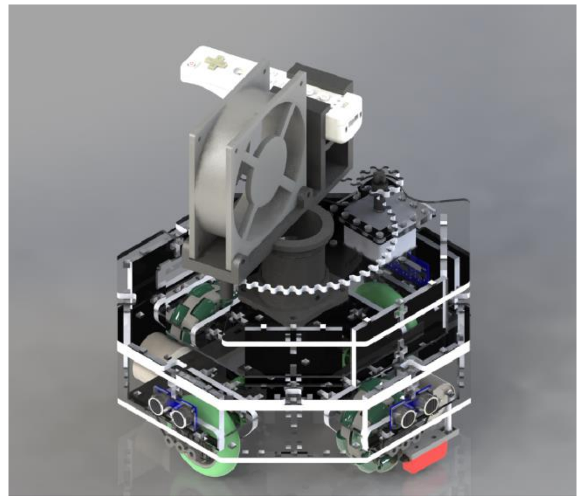


Figure 1: Rendered CAD Model

2.1 | Structure

Bearing in mind that a size limit was imposed on all teams, such that a team's robot could be no bigger than a box 12.5 inches to a measurable length or width, space was limited for the task at hand. With this in mind, the robot was designed to maximize space and to provide a modular approach without compromising functionality.

2.1.1 Drivetrain and Base

At the base of most every robot is its drivetrain; the way in which it is able to move about the field. Considering that there was simultaneously a time limit and a size restraint on the robot, a robot with a centralized virtual turning center was deemed more desirable, since potentially tight spaces on the field would be difficult to navigate otherwise. This ensures that minimal drift will

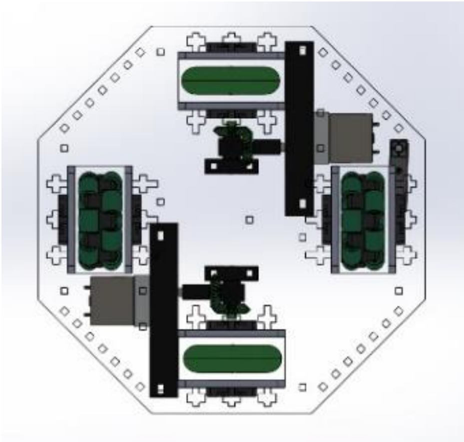


Figure 2: Robot Base

occur in the robot's position. In order to fit in these tight spaces, and thus not violate the size restriction, an octagonal base, as seen in figure 2, of 10.5 inches wide was designed, the shape of which would cascade into the upper sections of the robot. An octagonal base allows the robot to turn without having any corners protrude outward and collide with the walls. A circle could be argued to achieve the same objective whilst saving material cost, but the octagonal base allows the addition of any accessories to be at an equal and consistent angle around the base, hence keeping the robot's weight evenly distributed.

As seen in figure 1, four points of contact were maintained via the two traction and two omni-directional wheels, such that the robot would not tip at any one point during the challenge. The two traction wheels, being situated perpendicular to the pseudo-caster omni-wheels, were each directly powered by a Pololu 37Dx54L mm motor. This was accomplished using 45° beveled gears to divert the physical placement of the motor to the side. Doing so allows the motors available to be used while not interfering with each other. Each of the two traction wheels were directly driven by the aforementioned motors due to the fact that the field does not require any demanding motor output. Additionally, a speaker that played "The Raiders March"- better known as the Indiana Jones theme song- as well "We Didn't Start the Fire" by Billy Joel, was attached to the base as well to down fire towards the ground and amplify the sound as the soundwaves bounce off the ground.

2.1.2 Mid-section

Residing in between the drive train and the turret (top section), the mid-section, depicted in figure3, housed all the essentials for making the robot operate correctly. Maintaining the octagonal design, the mid-section was designed such that two distinct layers could be used to properly house and implement the necessary sensors and electronics. The first layer, which resided immediately above the drive train, primarily housed the IMU. Along with the IMU were the two motor controllers for the Pololu 37Dx54L motors. Additionally, the 12- and 5-volt batteries used to power the robot loosely resided in this section. Having these components on the bottom layer allowed for conserving space, neatness, and allowing the IMU to be centered on the robot.

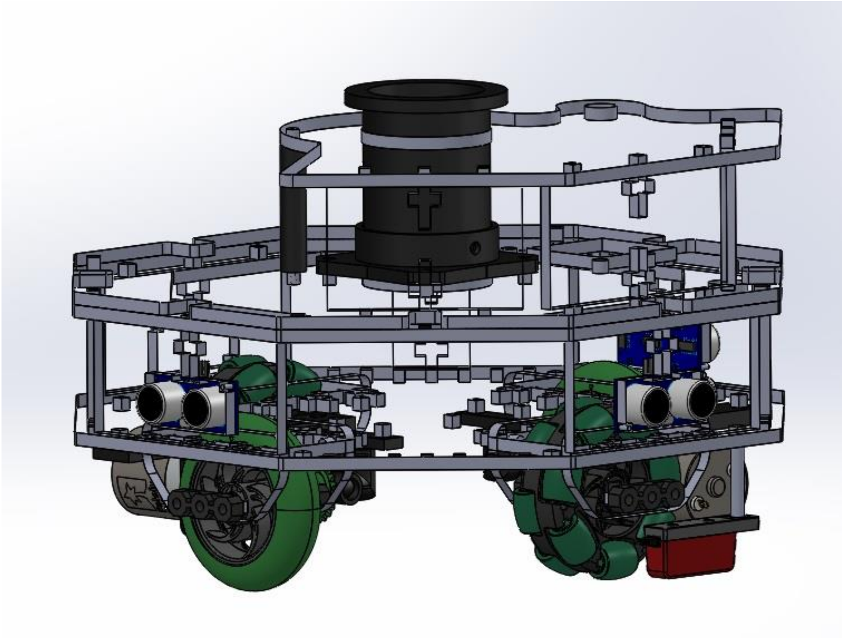
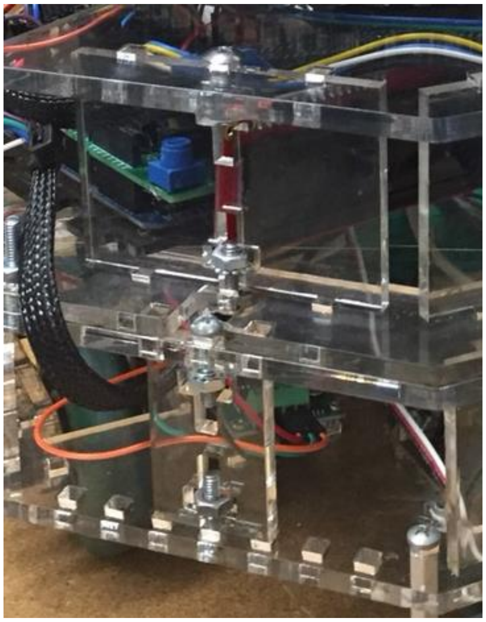


Figure 3: Robot Mid-section



Connecting the top layer of the mid-section were acrylic walls, as seen in figure 4. These walls served as standoff spacers to add stability and spacing for the wiring. This allowed the top layer to be connected only by a few screws. The spacers were positioned on the sides of the base that did not have any wheels. By placing the spacers in such a position, it allowed for the ultrasonic sensors to be attached to walls of acrylic above and parallel to the front, left, and right wheels separately. Having ultrasonic sensors positioned parallel to the wheels allow for accurate distance-from-wall measurements and straight driving from angle correction based on readings over time.

The top layer of the mid-section holds the majority of the electronics, that will be discussed in more detail later. In addition to the Arduino Mega and Uno, the Bluetooth shield, LCD screen and a communal breadboard all shared this space. It is in this area that the main computation and processing occurred.

2.1.3 Turret

Being the main asset of the robot, the turret, as seen in figure 5, was optimized to be free of unnecessary material. Centered on the concept of continuous scanning, a 200 cfm fan, Wii remote, and a Sharp Infrared Rangefinder were all assembled together on a 3D printed block that was mounted to a 100-toothed laser cut gear. This turret base was subsequently powered by a Mercury Nema Stepper Motor. Geared in a 10:1 ratio, the stepper motor's typical

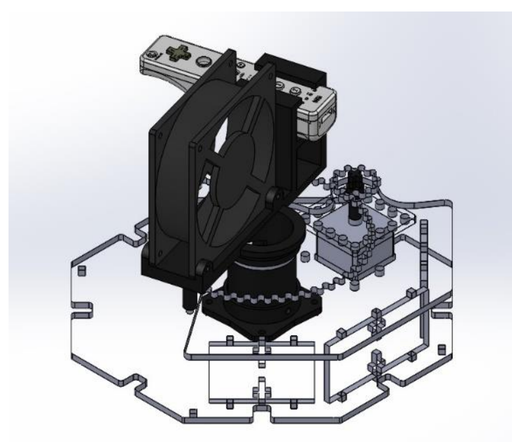


Figure 5: Robot Turret

1.8° per step was reduced to 0.18° per step, allowing a greater accuracy in determining the location of the flame with respect to the direction in which the robot was traveling. Care was taken to ensure that any and all wires deemed essential to the operation of the turret components where fed through the 3D printed base of the turret such that, while it was operating, the wires did not get tangled or break at extreme angles.

2.1.4 Modularity

The robot was designed modularly in order to get a base up and running as soon as possible. This way the turret and other sensors were cable of being calibrated and tested for functional operation separately and did not hinder the assembly of the base in order to get the navigation code working. This style of modularity also allowed the robot to be easily disassembled and debugged. The different sections can be disconnected and tested by the means of taking out only a few screws as opposed to taking the entire robot apart.

2.2 | Sensors and Electronics

In order to operate completely autonomously, as per the challenge rules, the designed robot needed to smoothly operate several sensors and electronic subsystems in order to optimize space and processing power. These systems were mounted throughout the robot chassis, as described in the previous section, in order to optimize their separate functionality.

2.2.1 The Fan

The most prominent system, the fan, was selected after careful experimentation. In compliance with the rules stated in appendix A, fire suppression methods such as fire extinguishers or snuffers were immediately disqualified in favor of air based methods. Considering that implementing compressed air would require an additional actuator, several fans were subsequently tested to determine which one would reliably extinguish the flame. After testing several smaller fans, it became clear that a relatively high volume of air would be required to extinguish the flame at any appreciable distance.

From this assumption, two options became clear. The robot could either implement a ducted fan or a larger fan combined with a focusing cone. Fans typically come in either 5- or 12-volt configurations, which presented an issue in terms of the amount of air the fan would blow in cubic feet per minute (CFM). Most 5-volt variants, as well as several 12-volt variants, did not supply sufficient air for their size requirements to be used in the robot. The solution presented itself in the form of the 12-volt fan seen in figure 6, which has a 200 CFM rating. In order to power the fan, a 12-volt battery was acquired and promptly wired into a relay on the communal breadboard, which in turn was activated via a digital output pin on the Arduino Mega.

Figure 6: 12-Volt 200 CFM Fan

2.2.2 Pololu Flame Sensor vs Wii Remote and Sharp IR

The next most prominent system in a successful robot was its ability to detect if there was a flame, and its location should it be present. A preliminary solution was seen in the Pololu flame sensor, seen in figure 7. This sensor operated by detecting infrared wavelengths in the range of 760 to 1100 nanometers, but its operable range was largely unknown. In order to test it, the flame sensor's analog values were recorded as it detected the candle at increasing distances. This data was then tabulated and graphed, as is seen in section 3.1. As a substitute to this, a Sharp IR Rangefinder was run through the same

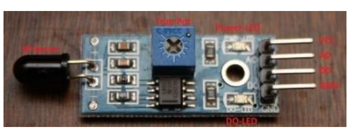


Figure 7: Pololu Flame Sensor

experiment, instead measuring the distance to a flat panel. After comparing the results of the two experiments, it was determined that utilizing the rangefinder would be more useful for locating the distance to the flame.

Figure 8: Wii Remote and Rangefinder

In regards to purely detecting the flame, the task was relegated to a standard Wii remote, as seen in Figure 8. In addition to being able to detect the flame, the Wii remote was capable of returning a horizontal and vertical angle within its approximately 30° field of view. These two angles could then be converted to a point on an xyz-coordinate system by interfacing with data acquired by the rangefinder, Mercury Nema Stepper motor, and Inertial Measurement unit. First, by finding the distance to the candle from the range finder and combining it with the angle calculated via the amount of steps the turret has turned from pointing straight forward, the Arduino can perform quick trigonometry to determine the planar xy-coordinates relative to the robot. From there, the absolute coordinates can be generated by interfacing with the global coordinate system calculated by the inertial measurement unit. The singular drawback to utilizing the Wii remote was the overhead required for communicating with it via Bluetooth, which was overcome by attaching a USB host shield for the Arduino Mega. Despite all this, the remote, in conjunction with the rangefinder, proved to be more reliable and easier to implement than the provided flame sensor.

2.2.3 Pololu IMU and Position Tracking

Nearly as important as knowing the location of the flame relative to the robot, is the robot's position relative to its starting point. In the context of a household firefighting robot, this is useful for knowing where a fire originated as well as returning to a starting position after successfully extinguishing a fire. This situation was tackled through the use of a provided 9-axis inertial measurement unit (IMU) and the digital encoders attached to the motors installed in the drivetrain.

The former sensor, the Pololu IMU, was used to determine rotational accuracy. Being secured to the center of the chassis, there was never a need to incorporate additional calculations in order to determine the total angle rotated. Originally, the data gathered by implementing solely the gyro was too prone to drifting, causing large discrepancies in between the calculated and real values turned. Therefore, the accelerometer was combined with the gyro in a complementary filter to significantly mitigate this issue.

The latter set of sensors, the digital encoders, were used to determine distance traveled in one of two perpendicular directions. As the motor axle turned, the encoders incremented the amount of ticks that passed. Using a simple conversion, these ticks were read as a linear distance, in either the X or Y directions. This grid, originally set by the IMU upon activation, was held constant throughout the robot's active operation. By utilizing these two sensors together, both the robot's and the candle's positions were easily maintained.

2.2.4 Processing and Communication

In order for every subsystem to function together, a centralized processing unit of some form was required. In the case of this robot, the provided Arduino Mega was initially selected. As an all-around decent microcontroller, the Mega possessed sufficient storage and RAM to handle the code that was uploaded to it. An additional Mega was used separately in order to play music, considering that the amount of memory needed for any individual song would prove too much when combined with the remainder of the code. In order to activate certain songs, the two boards communicated via I^2C protocol, with the provided Mega acting as the master and the music Mega acting as the slave. Additionally, the I^2C protocol was used to interface with the IMU directly, as well as the Bluetooth shield needed to communicate with the Wii remote.

Ultimately, a drawback presented itself in the board's processing speed. The microcontroller's multitasking capability proved sufficient for the task on hand. However, conditions may have been improved had the decision been made to use a multicore/multithreaded microcontroller, or even distributing some of the processing to a second microcontroller.

2.2.5 Miscellaneous Sensors

In addition to the sensors already covered, several key systems were essential in successfully extinguishing the candle. In order to detect when a cliff was present, a vex line sensor was implemented on the base of the robot, in front of the foremost Omni-wheel. Mainly unused, this sensor proved essential in preventing the robot from driving off the playing field.

As the primary sensor used in navigation, three ultrasonic sensors were installed in front and on the left and right of the robot. These sensors, which evaluate the time it takes for a sonic ping to travel from the emitter to the receiver, were used to maintain a safe distance from the walls to the right and in front of the robot, a topic covered more in section 2.3.1. By monitoring the distance reported over time, it was possible to ensure the robot was not deviating too much from travelling parallel to the wall.

Finally, an LCD screen, mounted within the mid-section of the robot, was used to debug the robot during tests as well as declare different states on the field during the final demo. The LCD would display where the robot was, if the flame was found, and ultimately where the robot calculated the flame to be. The particular function of the LCD is briefly convered in section 2.3.3.

2.3 | Programming

On an overarching scale, the robot followed the cascading state machine exhibited in figure 9 below.

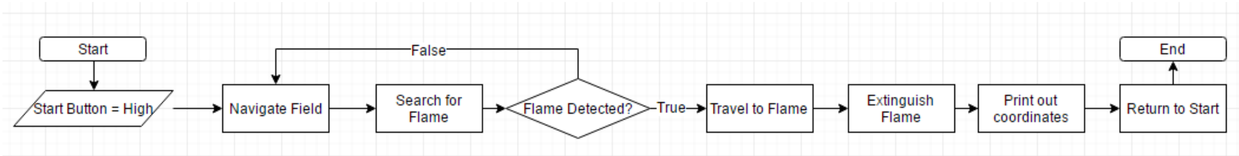


Figure 9: Overhead Programming Flowchart

The robot was designed to run two simultaneous state machines in order to hit the design goal of continuous scanning while moving. The first, and arguably more important, is the navigation state machine, which utilizes input from the various sensors in the mid-section as well as the turret control state machine to determine which state to jump to next. The turret control state machine, on the other hand, receives input from the Wii remote to determine how may steps the stepper motor should take.

2.3.1 Navigation

The most essential state machine of any mobile robot, the navigation code, seen in figure 10, allows the robot to adapt to any situation it found itself in. The code began by checking its position relative to the walls, if any, via the ultrasonic sensors. Based on this feedback, the robot followed a set of states that ended with the it parallel to a wall on the right side. Once this situation was established, the robot remained in the primary state- dubbed DriveForwardWall.

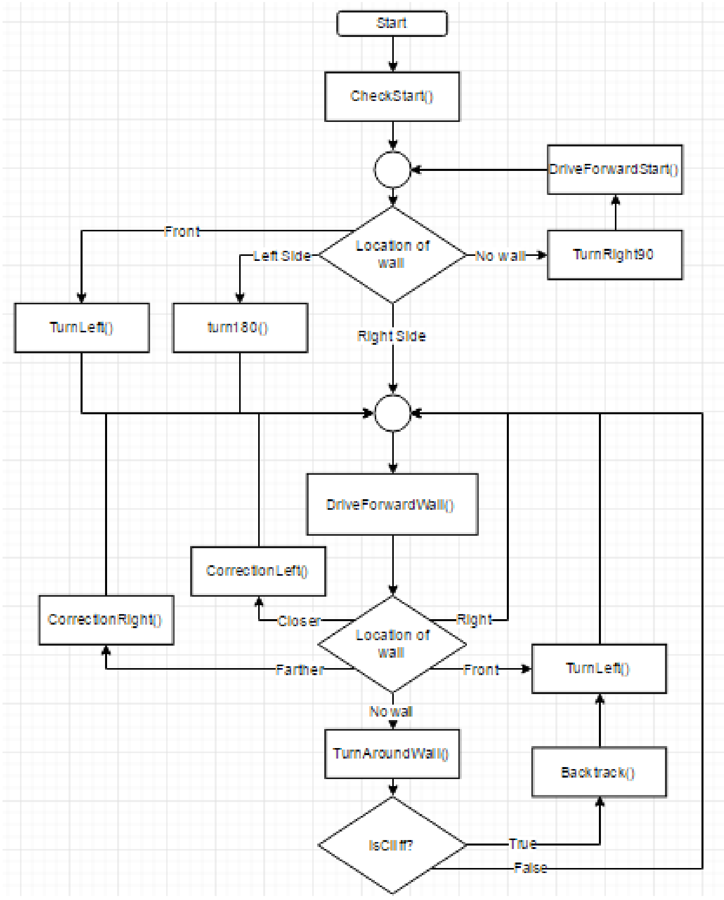


Figure 10: Navigation Flow Chart

This state routinely checked the status of the robot with respect to the presence, or lack thereof, of a wall in its immediate vicinity. Using the ultrasonic sensors, the robot checked if a wall was within 15 cm of the front of the chassis as well as if the robot was drifting too far from or too close to the wall on the right. If a wall was detected in front of the robot, a counterclockwise 90-degree turn was performed, using the IMU as mentioned in section 2.2.3. If the robot found itself drifting too far from or close to the wall, a slight turn of 3 degrees in the direction needed was performed in order to become parallel with said wall again. If the robot is severely tilted, the robot continued to try to correct itself in 3-degree increments. After successfully completing any singular function, the robot would then revert back to the primary state, DriveForwardWall.

If the robot suddenly did not sense a wall on the right side it entered a subset of states that determined if there was a cliff or if the robot should make a U-turn. Using the encoders, the robot drove forward 6 inches in order to ensure it cleared the wall and then performed a 90-degree clockwise turn. It saved its position after turning for reference later. The robot then drove forward until it detected a cliff or it saw a wall on the right again. If a cliff was detected with the line follower sensor the robot backed up until it was back at the saved position and then turned 90-degrees counterclockwise. It would then continue to drive forward until it found another wall on the right and subsequently switch back to the primary state. If there is not a cliff the robot will detect a wall on the right side. It drove forward 9 inches, then turned right 90 degrees. It drove forward until it detected the other side of the wall on its right side and then followed it or if there was a wall in front of it after the turn, it turned left 90 degrees to follow that wall. This allowed the robot to navigate the maze in any configuration.

It would continue through these states indefinitely, keeping track of its position with the encoders, until the turret control detected a flame within a certain distance to the robot. The navigation code was used again to return the robot to the starting position except it would stop once it is within a certain tolerance from the 0,0 coordinates of the table.

The complete navigation code can be found in the code submission box.

2.3.2 Turret Control

As mentioned prior, the turret control state machine relied upon the Wii remote's ability to detect the candle flame. As seen in figure 11, the method employed two main states FLAME_SEEN and NO_FLAME_SEEN, which were for when the Wii remote detected a grouping of infrared and when it did not. Both states relied on were subjected to several conditional statements to consistently test the status of the flame within the field of view of the Wii remote. If the flame was not seen by the remote, the stepper motor would continue to take ten steps at a time in the same direction until it reached a maximum value. Since the robot followed the right wall due to the navigation code, the turret would only need to scan to the left of the robot. This meant that the maximum values would only need to be 0 and ½ of a turret revolution, in the direction of travel and in the reverse direction respectively.

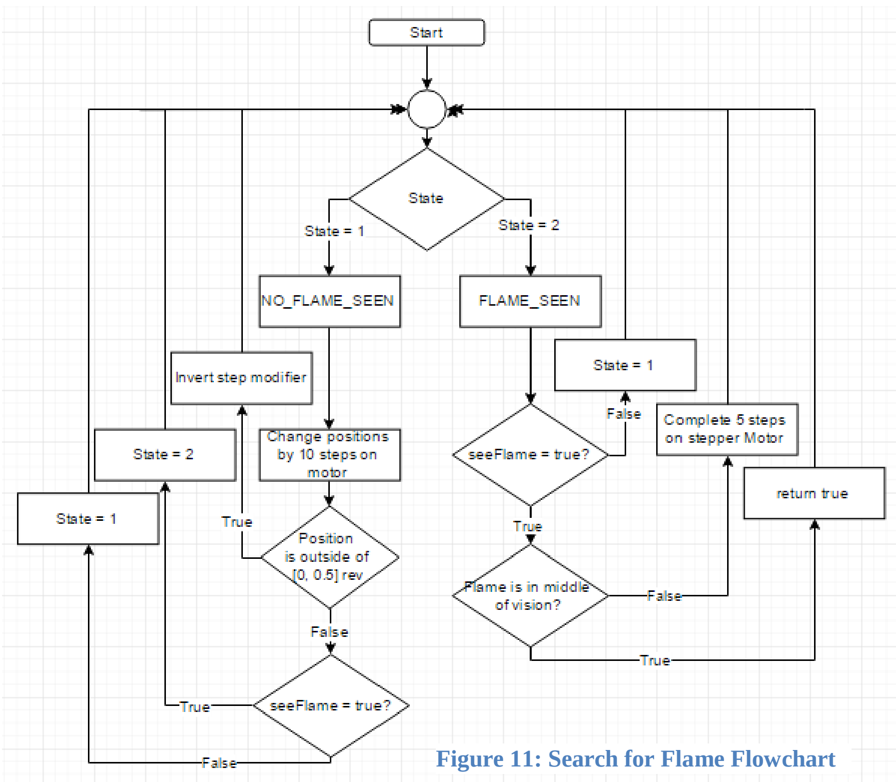


Figure 11: Search for Flame Flowchart

If, at any point, the Wii remote picked up the candle, the state would immediately switch to FLAME_SEEN. In this state, the Mercury Stepper would take five steps in the direction it was just moving such that the spot of infrared was placed in the center of the Wii remote's field of view, upon which the method would return true to the overhead code, activating a response there. However, if the candle once more passed out of the remote's FOV, the method state would revert back to NO_FLAME_SEEN. The direction in which it turns was decided by a global variable, “mod,” which takes the values of -1 and 1, for clockwise and counterclockwise rotation respectively, that is set in the former state, but carries over to the latter once the flame is seen.

The turret code also modified the value isCentered, for when the flame was within a small area in the center of the Wiimote’s view. The Wiimote also returned a flame size. These were used by the overhead code to determine when to turn on the fan and extinguish the flame. If the flame size was at least a certain minimum value, then the overhead code stopped the robot from entering the navigation state machine and stop the robot’s movement. The turret control then can center the Wiimote onto the flame, changing isCentered to true. The overhead code would turn on the fan to put out the flame and stop the turret control state machine. After a period of about 4 seconds the robot would reactivate the turret control to verify the flame was successfully extinguished. If it hadn’t, the fan will continue to blow until it was successful. Once extinguished, the fan was turned off and the robot reentered the navigation state machine to return to start.

The complete turret code can be found in the code submission box.

2.3.3 Other Code

The robot also had several other features using the LCD screen and IR range finder. The robot position in X, Y and Z (Z is kept 0), were displayed on the LCD while it was searching for the flame. When the turret control had not sensed a flame, “NO FIRE” was printed on the LCD display above the robot position. Once a fire was found, this was replaced by “FIRE FOUND!”. With the flame extinguished, the IR range finder determined the distance from the robot to the candle, which with some trigonometry returned the X and Y distances the candle is from the robot. Using the robot’s direction, which was saved in a variable, these distances were added to the robot’s global position, yielding the candle’s absolute position in X and Y. The Z component was determined via the Wiimote implementing more trig and the distance returned from the rangefinder from the robot to the candle. “Candle Position” then replaced “FIRE FOUND!” on the LCD display, while the Candle Position replaced the Robot Position.

The complete robot code can be found in the code submission box containing these functions.

Section 3: Results and Discussion

3.1 | Sensor Results

Since autonomy is paramount to the operational success of the robot in this challenge, each sensor utilized should be tested to thoroughly understand its individual behavior in response to different situations. As described in section 2.2, the Pololu Flame Sensor was evaluated to determine its effectiveness in discerning the distance a flame was from the robot. The results from this similar experiment can be seen in table 1 and figure 12.

Distance (inches)	Flame IR Value
3	13
6	16
9	19
12	22
15	24
18	28
21	30
24	35
27	35
30	42
33	52
36	75
39	113
42	141

45	170
48	220

Table 1: Flame Sensor Distance Evaluation

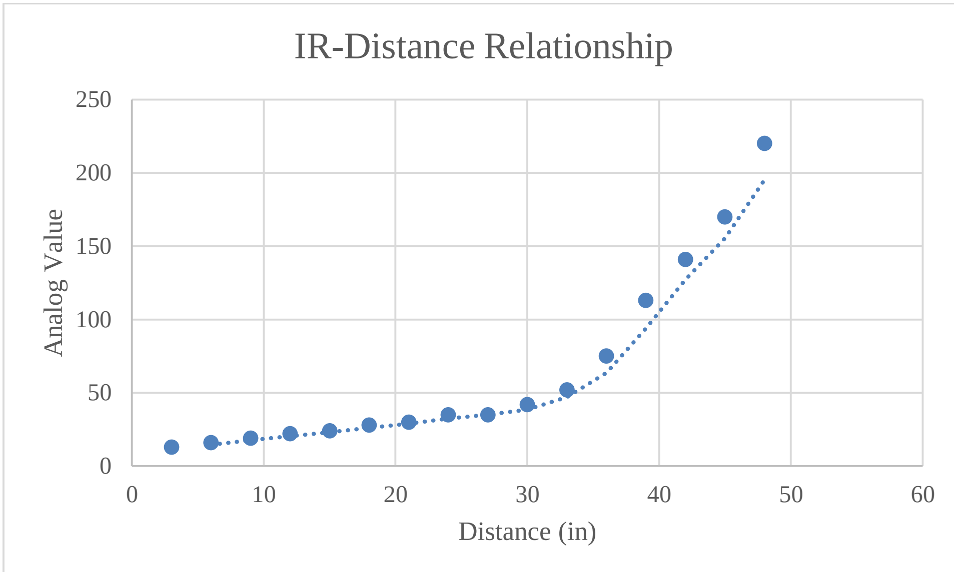


Figure 12: Distance vs Analog Value

Additionally, as described in the aforementioned section, the Sharp Infrared Range finder was similarly evaluated for use in determining the linear distance from the flame to the robot. The tabular and graphical results from evaluating the sensor can be seen in table 2 and figure 13 respectively.

Distance (inches)	Rangefinder Value
6	340
12	171
18	120
24	93
30	75
36	71
42	59
48	58

Table 2: Range Finder Distance Evaluation

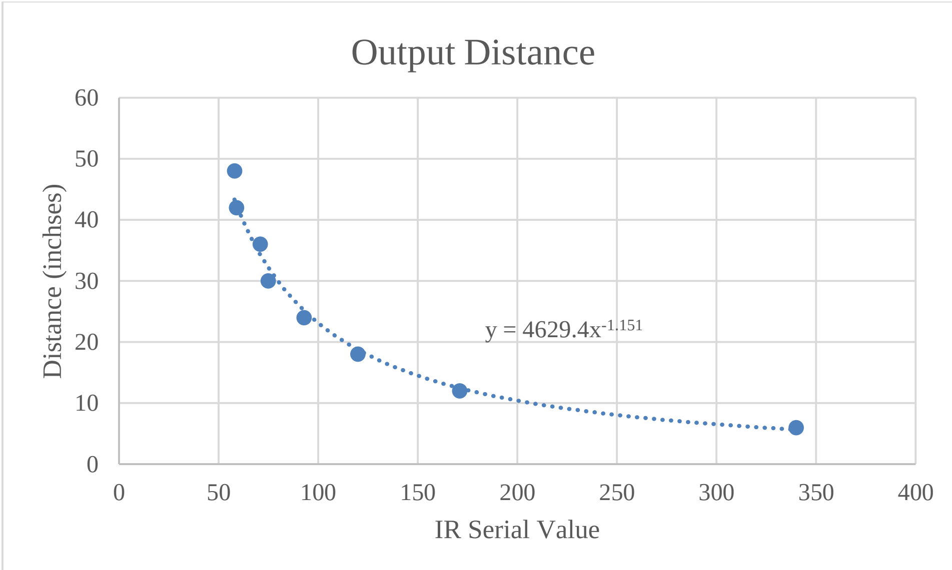


Figure 13: Serial Value vs Distance

3.1.1 Discussion

With the Sharp IR Rangefinder and the Pololu flame sensor, it was necessary to determine their respective reliability in determining the distance to the flame. While the two sensors both utilized IR to identify distance, it became clear that the Sharp Range finder was more reliable. As seen in figure _ (FS graph), the Pololu flame sensor exhibited a linear behavior of a particular slope up until approximately 33 inches, after which it assumed a new slope. In addition to the difficulty "seeing" the flame, the flame sensor was deemed too difficult to implement into the main programming, considering no singular equation could be used diagram the behavior of the sensor.

Contrary to this, the Sharp Rangefinder exhibited a consistent behavior over the course the test, as seen in figure_. Relatively exponential in nature, the rangefinder showed some difficulty with evaluating distances within 6 inches, and was very sensitive to distance changes past 42 inches. However, since these values appeared to follow a consistent trend, Excel was used to generate the standard equation below to map the incoming analog values to a distance. This feature was implemented directly into the navigation code as a way to check that the flame was within the operable range of the fan.

3.2 | Presentation Day Results

On presentation day, the demo field was rearranged and set up in the style depicted in the drawing seen in figure 14. The solid black lines are the walls that stand approximately 13-inches high. The red and blue circles represent the starting locations of the candle and robot respectively on each of the two attempts, with “R1” and “C1” corresponding to the first attempt and “R2” and “C2” corresponding to the second attempt.

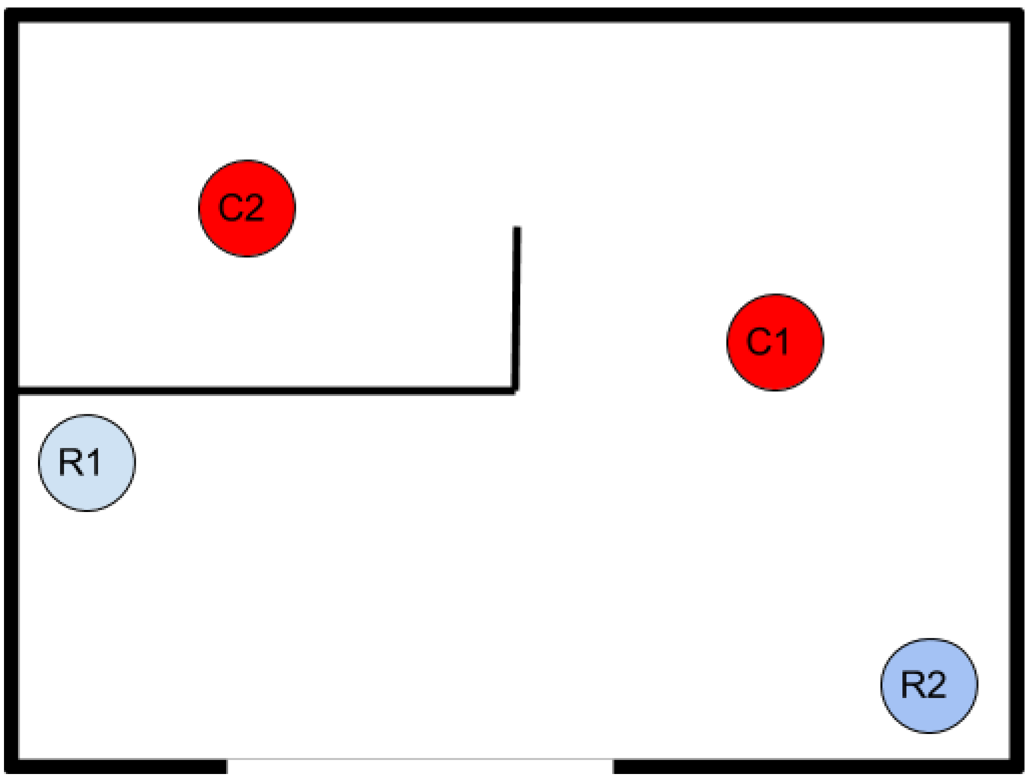


Figure 14: Demo Day Setup

In the first attempt, the robot failed. As it drove down from R1 toward the lower left corner with respect to the above image, the robot followed the wall on its right until it detected the candle at C1. It then proceeded to turn on the fan for approximately 4 seconds, after which it resumed normal navigation. The robot successfully navigated the cliff, located at the bottom of the above image, and the entirety of the field, but failed to stop once it reached the general area around R1.

In the second attempt, the robot once again failed. Starting at R2, it proceeded to drive up towards the upper right corner, before detecting the flame about halfway up the side. After failing once more to blow out the flame, considering there was a wall in between the fan and the candle, the robot's operation was terminated.

3.2.1 Discussion

The presentation day results were better than expected but the robot still ultimately failed to put out the flame. The navigation code ran great as the robot moved around the field perfectly and accurately kept its position. The turret control worked better than expected, detecting the flame from across the field. It was not expected to activate the fan until it was within a few feet from the flame, however that line of code was never encountered. The visible result was the robot trying to extinguish the flame at the maximum radius of the fan and failed to completely blow out the flame, allowing for it to spring back to life. Putting out the flame manually allowed the robot to return to its original position.

The problem with stopping once the robot had returned to the R1 vicinity was that the navigation code did not have instructions on how to turn around a wall on the outside. This scenario was unforeseen as the test fields never had this so it was never programmed into the robot. The visible result was that the robot drove into the wall instead of stopping. The robot was positioned correctly and allowed to continue movement but because it thought it was moving in a direction it wasn't the robot never stopped at the starting position. These problems could easily be solved with a few minor additions to the code so overall the robot performed well.

3.3 | Video Resubmission Results

As seen in the video submitted to the online Dropbox, the robot was ultimately successful in accomplishing the primary tasks. In replicating the first attempt, the robot completed navigation in the same fashion seen on the official demo day, navigating along its right wall and avoiding falling off the cliff. In this attempt, however, it was able to blow out the candle completely, as it only activated the fan once the robot was essentially immediately next to the candle stand. Additionally, the video showcases the robot's ability to display the robot's as well as the candle's position relative to its starting point.

3.3.1 Discussion

The results of the video were much better than the presentation. The robot could fully navigate the field, adding instructions for the scenario described in section 3.2.1 above, and the robot waited to be a certain distance away from the flame before trying to put it out. The result was that the flame was put out completely.

However, the robot still failed to return to the original position. The main reason appears to be that the field was in poor shape after the demos took place. The tables were at slight angles, making it difficult to turn accurately at certain positions on the field. The robot was fine following the outside wall but navigating around the spiral of the maze was too difficult and it kept driving off course. The return to start function was not necessary for full credit so in the interest of time this function was never improved on.

3.4 | Future Improvements

After reviewing the robot's performance in both the official demonstrations as well as the resubmission videos, it was determined that steps needed to be taken in three key areas.

First, the essential programming needs to be dispersed from one Arduino Mega to two or more, communicating in I^2C, such that the program(s) can be run at faster rates than if all the processing was done on a singular board. In the case of this robot, it was found that only about 20 complete runs of the main loop were completed each second. This was found to be too slow, and mainly due to the fact that programs for three separate boards were integrated onto a singular microcontroller.

Second, the programming must be refined to incorporate the redundancy of the ultrasonic sensors, encoders, and inertial measurement unit to ensure straight line travel and accurate coordinates as well as to eliminate any drifting that might occur while driving. In lieu of relying solely on the ultrasonic sensors to determine straight line travel, the encoders on each of the two Pololu motors should be consulted as well to ensure that each wheel is going through the same number of revolutions. Alternatively, a different form of navigation could be implemented that utilizes a form of computer vision to map the immediate area, breaking the reliance on the slower wall following technique.

Finally, steps must be taken to reduce the cost and overall cleanliness of the robot in order for it to be considered a household commodity. As seen in appendix B, the total cost to produce a singular robot following this report is approximately $510. While this cost may be reduced somewhat when producing in bulk, it is not a significant difference. An iRobot Roomba, which cleans an entire household, is valued at around $400, nearly a hundred dollars less than this prototype. If this robot is to follow the standard set, of being a household commodity, the price needs to be drastically reduced, as well as being designed to encompass all necessary components, especially wires. As seen in figure 15, wires spill out of every appreciable side of the robot, giving

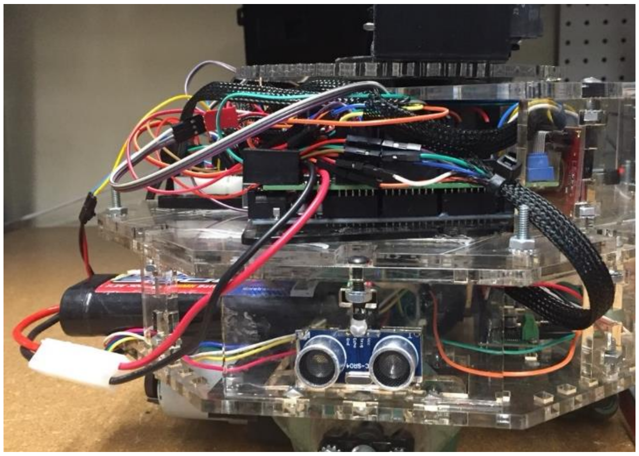


Figure 15: Disorganized Wires

it an unkempt, messy look, as well as pose a risk to itself and those around it. If these could be reorganized and contained within a singular body, the robot would be able to be integrated into homes everywhere without fear of electrocution or accidentally pulling out an essential wire.

Section 4: Conclusion

In an effort to simulate a firefighting version of the iRobot Roomba, the robot was able to autonomously navigate and locate the flame, yet initially failed at the ultimate goal of extinguishing the flame completely on its first try, eventually succeeding with some last minute changes to the code. By implementing a Wii remote and an infrared rangefinder, along with ultrasonic sensors and an IMU, the robot was able to navigate, find, and extinguish the flame with the use of a 12-volt, 200 CFM fan. Steps were taken to implement several additional microcontroller boards as well as necessary shields to allow certain features such as Bluetooth communication with the Wiimote and music to exist. While ultimately successful, the robot could still do with additional work in programming refinement and distribution, as well as efforts to reduce its cost and improve its safety appeal, in an effort to make it more suitable for an average American home.

References

[1]"What chemicals are used in a fire extinguisher? How do they work to put out fires?", *Scientific American*, 2017. [Online]. Available: https://www.scientificamerican.com/article/what-chemicals-are-used-i/. [Accessed: 01- May- 2017].

[2]"Home Fires | Ready.gov", *Ready.gov*, 2016. [Online]. Available: https://www.ready.gov/home-fires. [Accessed: 01- May- 2017].

[3]M. Ahrens, "NFPA report - Home Structure Fires", *Nfpa.org*, 2016. [Online]. Available: http://www.nfpa.org/news-and-research/fire-statistics-and-reports/fire-statistics/fires-by-property-type/residential/home-structure-fires. [Accessed: 01- May- 2017].

[4]"NFPA statistics - Home fires", *Nfpa.org*, 2016. [Online]. Available: http://www.nfpa.org/news-and-research/fire-statistics-and-reports/fire-statistics/fires-by-property-type/residential/home-fires. [Accessed: 01- May- 2017].

[5]C. Putnam, *Lab Assignment 5: Firefighting Challenge*, 1st ed. Worcester, MA: WPI, 2017, pp. 1-2.

Appendix A: Final Project Requirements

INTRODUCTION

In this term-long lab assignment you are required to build a robot capable of navigating a structured environment, finding and extinguishing the simulated fire (a candle), and report the location of the fire in reference to where the robot started. As in RBE 2001, your team will be provided with a Kit of Parts (KOP) which will include an Arduino Mega 2560 and various sensors including a specialized flame sensor.

GOALS

1. Safely navigate the field while avoiding any obstacles that may be found.
2. Find the fire and indicate that the fire has been found.
3. Extinguish the flame and indicate that the fire has been extinguished.
4. Report back the estimated location (X, Y) of the flame using the KOP IMU, encoders, or any other strategy approved by the professor or course staff.
5. The robot must use at least the gyro (part of the IMU), encoders, and a range sensor. Additional points may be awarded for the effective implementation of additional sensors.

PLAYING FIELD

The playing field will consist of a series of walls, cliffs, and obstacles. The approximate size of the field will be 8' x 12'. The candle flame will range from 10 cm to 30 cm above the nominal playing field level. The playing field is <u>not guaranteed</u> to be fully enclosed.

REQUIREMENTS

1. The task must be completed autonomously.
2. The robot should be no larger than a 12.5" cube at the start of the challenge.
3. The robot must indicate when the fire is found and when the fire has been extinguished. [LED's are recommended but not required]
4. The robot must output an X, Y estimate of the location of the fire in inches from the starting location. [LCD screen recommended but not required]
5. In the final report, explain the use and operation of the IR flame sensor. Show a plot of its effective range derived from your calibration data.
6. The robot must complete the task within 5 minutes including the return to the starting position if that is attempted (see optional item 8 below).
7. Optional / Extra Credit: The robot reports the estimated Z location of the fire (in inches from the playing surface).
8. Optional / Extra Credit: The robot returns to its starting position.

RESTRICTIONS

1. The robot must be designed in such a way that it does not combust if the robot comes into contact with the candle.
2. No combustible fluids or gasses may be used. [For example, extinguishing the flame with a flamethrower is unacceptable.]

3. The robot must fit within a 13” x 13” x 13” box at the start of the challenge without touching the walls or top of the box. In other words, the box may not be used to constrain the robot in any way.
4. The robot must remain in contact with the floor of the playing field at all times. Translation: no flying robots!!
5. The robot must operate fully autonomously. The robot may be powered on after it has been positioned on the playing field. It should not begin any driving operations until after a “Go” button has been pressed.
6. The use of water or other liquids is not recommended but is not against the rules. With that said, if you use liquids and there is any damage to electrical components provided to you in the KOP (Arduino, sensors, etc.), your group will be financially responsible for replacing the parts regardless of what happened.
7. Contact with a wall is not allowed. Touching the candle or the candle-holder is not allowed. Contact any of the obstacles on the field is not allowed.
8. If a cliff is present (e.g., if a section of outside wall is missing), the cliff will be marked in such a way that it can be detected by a suitably equipped robot in time to avoid driving over the cliff.

POST LAB

1. Write an excellent lab report including the following sections: Abstract, Table of Contents, Introduction, Methodology, Analysis, Results, Discussion, and Conclusion. Include in your report a cost analysis of your robot, an analysis of the driveline, and an electrical systems analysis.
2. Explain the use and operation of the IR flame sensor and plot its effective range (derived from your calibration data).
3. Describe and discuss any testing done with other sensors used to complete the task. Document how well they worked, problems you had using them, etc.
4. Describe and discuss the mechanism used to extinguish the fire along with any relevant design decisions.
5. Demonstrate good software development and coding practices.
6. Fully document your software procedures, problems encountered and solved, special techniques used, and so forth.

Appendix B: In Depth Cost Assessment

	Item	Quantity	Price	Total
Base	Line Tracker (3 pk)	1	$ 39.99	$ 39.99
	2.75" Omni-Directional Wheel Double Roller (2 pk)	1	$ 19.99	$ 19.99
	2.75" Wheel (4 pk)	1	$ 9.99	$ 9.99
	Metal Gearmotor 37Dx54L mm with 64 CPR Encoder (50:1)	2	$ 34.95	$ 69.90
	Total			$139.87
Mid-section	HC-SR04 Ultrasonic Sensor Distance Module (5pcs) for Arduino UNO MEGA2560 Nano Robot XBee ZigBee by ElecRight	1	$ 9.79	$ 9.79
	MinIMU-9 v3 Gyro, Accelerometer, and Compass	1	$ 11.95	$ 11.95
	MC33926 Motor Driver Carrier	2	$ 17.95	$ 35.90
	Arduino USB Host Shield	1	$ 12.99	$ 12.99
	Arduino Uno Rev 3 with Long Pins	1	$ 24.95	$ 24.95
	Arduino Mega 2560 Rev 3	1	$ 45.95	$ 45.95
	Vex Limit Switch (2 pk)	1	$ 12.99	$ 12.99
	5/16" PET Cable Wire Wrap Expandable Braided Sleeving 8M 26ft	1	$ 7.62	$ 7.62
	helishun hls-14f3l-dc12v-c relay	1	$ 1.00	$ 1.00
	Total			$163.14
Top	Mercury Nema Stepper Motor SM-42BYG011-25	1	$ 17.97	$ 17.97
	Wii Remote Plus	1	$ 39.99	$ 39.99
	Ultra Strong 12cm DC 12V 200 CFM Fan	1	$ 19.99	$ 19.99
	Sharp GP2Y0A21YK0F Analog Distance Sensor 10-80cm	1	$ 9.95	$ 9.95
	Total			$ 87.90
Misc	5 mm Acrylic Sheet	3	$ 21.13	$ 63.39
	0.22" Acylic Sheet	1	$ 19.99	$ 19.99
	Elegoo 120pcs Multicolored Dupont Wire 40pin Male to Female, 40pin Male to Male, 40pin Female to Female Breadboard Jumper Wires Ribbon Cables Kit for arduino	1	$ 8.86	$ 8.86
	Total			$ 92.24
	Grand Total before tax			$483.15
	Grand Total after tax			$513.35
	Personal Contribution Total			$287.42

Appendix C: Authorship & Work Distribution

Section	Author
Introduction	Ben Wagner
Methodology	Team 12
Results	Nick Lanotte, Ben Wagner
Conclusion	Ben Wagner

Team Member	Percentage	Work Contribution
Dillon Arnold	24	Significant digital and physical development of robot Assisted in completely wiring everything Researched necessary parts (i.e. Fan) Spearheaded 3D printing properties
Parker Grant	24	Provided solution to flame sensing Some programming development, support (Turret Control), and debugging (Turret Control & Navigation) Electrical problem solving
Nick Lanotte	28	Significant digital and physical development of robot Significant robot programming and debugging (i.e. 80% of it) Electrical problem solving
Ben Wagner	24	Acquisition of materials and parts Adequate programming development (Turret control) Power Point development Final Report Writing Team Management