Ernie Patenia, Samuel Stephen, Victor Brzeski, and Fergyl Dennis Estioco

IEEE Robotics Team 2

December 9th, 2014

Dr. Leonard P. Trombetta, PhD

Cullen College of Engineering,

4800 Calhoun Road,

Houston Texas, 77004

Dear Dr. Trombetta,

Enclosed is the Final Technical Report which describes the final status of our work completed during the Fall 2014 semester. This is intended to enumerate our accomplishments in technical detail to assure yourself, Dr. Marpaung, and Dr. Glover that our team was on schedule and has progressed through the project in a timely manner that will ensure completion by April 2015 for the IEEE R5 Robotics Competition.

This report talks in technical detail about each of the three modules and what has been accomplished for each module. Following completed objectives, future plans are discussed which will ensure that we continue to progress on schedule into the Spring 2015 semester.

If you have any questions, comments or concerns regarding this report or on the general progress of our project, please do not hesitate to contact me, Ernest Patenia at 281-750-2116 or at elpatenia@uh.edu.

Thanking You

Sincerely,

Ernest Patenia

Basically this is kind of brief. There is a lot missing in terms of technical description of the components you’re using as well as of the design contest itself. The description of current progress is minimal. What tests did you do? How did they come out? There is very little about this. More detailed comments below…IEEE R5 2015 ROBOTICS COMPETITION

FINAL TECHNICAL REPORT

**Submitted on:**

December 9, 2014

**Submitted by:**  
*IEEE Robotics Team 2*

Ernie Patenia

Samuel Stephen

Fergyl Dennis Estioco

Victor Brzeski

**Sponsored by:**  
Dr. John Glover

Abstract

In order to participate in the IEEE R5 2015 Robotics Competition, IEEE Robotics Team 2 must build a robot with an emphasis on robustness, that is, a robot that can handle non-idealities that may arise during the actual competition as opposed to having an ideal track that conforms to the announced maze board specifications. The competition requires the robot to traverse a maze and visit every square, identify ASCII characters that may appear on surrounding walls, and identify the optimal path solution that links the beginning of the maze to the end. In order to realize our target objective of accomplishing these competition tasks, we have divided our robot’s system into four modules: Locomotion (the physical movement of the robot chassis throughout the maze), camera motion (the movement of the camera servo mount to allow the Raspberry Pi camera to take pictures), image processing (the recognition of ASCII characters based on pictures taken from the Raspberry Pi camera module), and maze navigation (the state machine of the robot as it moves through the maze, ensuring that it eventually solves the maze and keeps track of its current position within the maze). As of December 9th 2014, the robot can move in all cardinal directions, strafe left and right, and rotate on its axis. Software has also been written to optically recognize a string of ASCII characters from 6 inches away, and a maze navigation algorithm has been written to solve user-defined mazes.

1. **Background and Goal**

This year’s 2015 IEEE Region 5 Robotics Competition is based on a maze. The competition consists of both a search phase and a critical path phase. During the search phase, characters will also need optical recognition which adds a layer of difficulty. As this is a prestigious competition, it would be very beneficial for our school to place, and hopefully win, the competition to increase the Cullen College of Engineering’s reputation at the University of Houston and nationwide. Below in Figure 1 is an image of a sample maze provided by IEEE.



Figure : 2015 IEEE Robotics Competition Track [1]

Our goal is to place within the top 3 finalists of the competition. In order to achieve this goal, we must design and build our robot in a robust way which can handle potential deviations of the competition specifications. To realize a robust design, we have split up our design into three practical modules: Image Processing, Maze Navigation, and Locomotion. This modular design allows us to optimize individual parts of our design without interfering with other parts.

1. **Problem, Need, and Significance**

The University of Houston has not won the IEEE Region 5 Robotics Competition in many years. Winning the competition would increase the University of Houston’s recognition among other Region 5 which would inspire prospective students to attend the Electrical and Computer Engineering College. This recognition would also bring more funding for the college which will help assist in providing students with more learning tools and equipment.

To solve this problem, our team needs to build a robustly designed robot which can handle deviations of the competition specifications. This is key to success as if any part of the competition is not ideal, our robot may not succeed. This applies to how consistently we can design locomotion, how consistently we can recognize the character Easter eggs, and how robust we can design the maze navigation algorithm.

This type of competition brings forth many embedded systems and electrical concepts that are learned throughout an undergraduates studies at the University of Houston. Because of this, the design of this robot will help to inspire and teach future students of these concepts.

1. **User Analysis**

The design of this robot which will compete in the IEEE 2015 Region 5 Robotics Competition has to be modular, but in this design, our team wants to consider other users who may have use of the robot. Our team has considered other potential users of our robot, including other Electrical and Computer Engineering students along with potential incoming freshmen to show them an example of what they could accomplish at the University of Houston.

1. **Overview Diagram**

Our modularly designed robot can be seen below in Figure 2.

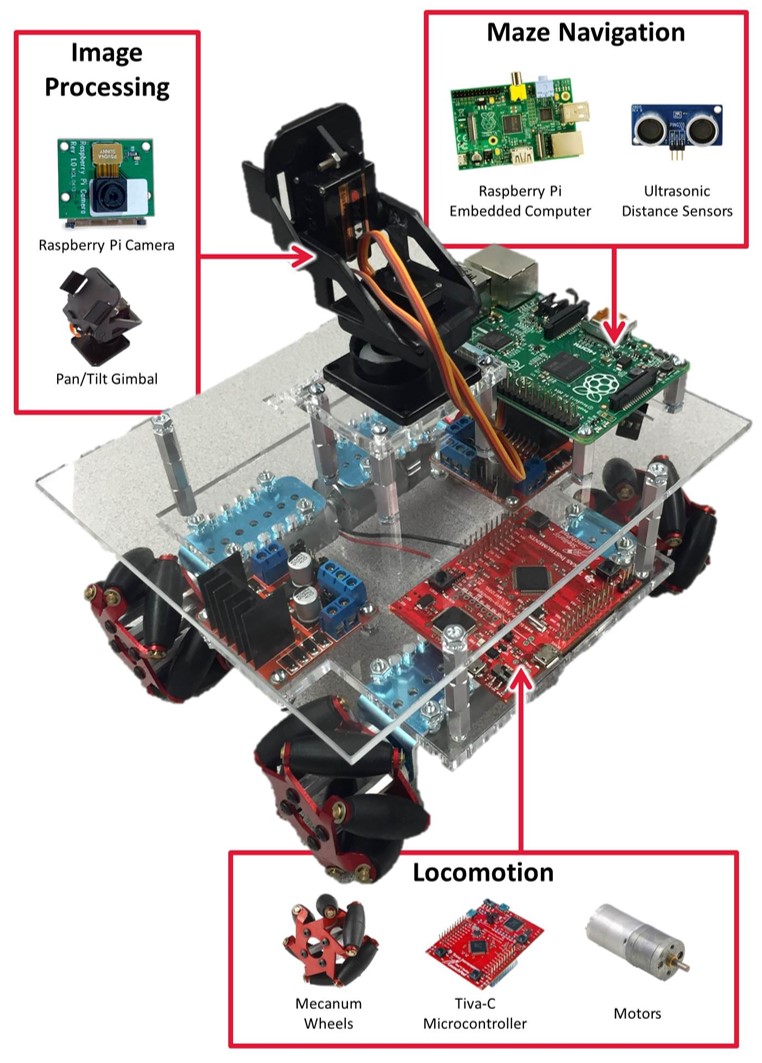


Figure : Overview diagram for the robot design. The Raspberry Pi [2] manages the maze solving algorithm [3], the camera module [4], the distance sensor [5], and overall logic. The TivaC [6] manages the Mecanum wheels [7] and the accompanying motors [8] and pan/tilt gimbal [9]

The modules in our design are Image Processing, Maze Navigation, and Locomotion. These modules allow us to independently design, build, and manage different parts of our robot without interfering with other aspects of our robot.

The Image Processing module includes a Raspberry Pi, a Raspberry Pi Camera Module, and a Pan/Tilt Gimbal. These elements together will work together to take images of the front, left, and right walls once our robot enters a square cell. Once these images are taken, they are then queued to be processed.

The Maze Navigation module includes the Raspberry Pi, the ultrasonic sensors, and potentially line array sensors. The Raspberry Pi is to be the brains of the robot, therefore it will handle the logical decisions of the robot. Once all modules have been build, tested, and combined, the Raspberry Pi will communicate with the Tiva-C to get the motors and servos to move. In this approach, the Tiva-C will be doing all the heavy lifting, while the Raspberry Pi only has to worry about logical decisions.

The Locomotion module includes the Tiva-C and the motors and servos that it will control. As stated previously, the Raspberry Pi will communicate instructions to the Tiva-C on which direction to move or which servo to rotate. This is key in allowing the Raspberry Pi to do more CPU intensive instructions without having to worry about the motors/servos.

1. **Target Objective and Goal Analysis**

Our Target Objective for this entire project is to be able to complete the maze of the competition and also score at least 85% of the maximum possible points. Completing this target objective means to be able to run any configured maze successfully within 5 minutes while correctly recognizing at least 85% of the characters on the walls. To accomplish this goal, each of the modules is divided into specific objectives that would confirm progress of its enclosed module. The goal analysis which encompasses these objectives can be see below in Figure 3.



Figure 3: Goal Analysis with Target Objective for Spring 2015

The primary module is the Maze Navigation Module. This module is divided into two primary objectives: to traverse the maze successfully and to calculate the critical path of the maze successfully. This is one of the more important modules as without a maze navigation algorithm, we will not be able to traverse the maze during the competition at any capacity. Currently, the maze navigation algorithm is operational but not perfect in simulation. Currently, the algorithm does not run the optimal path to enter every square of the maze, but simple takes an excess of decisions to complete the maze. This algorithm was implemented using C in a Linux-based environment as this is what the Raspberry Pi uses. During runtime of the program, each square move is displayed in order using the numbers of each square as reference. AT the conclusion of the program, statistics of the run are given, including number of rotations, moves, and number of unique squares visited. Continuing through into next semester, the maze algorithm will be modified to be more robust and to be able to run the most efficient path for both phases of the competition.

The other module that directly relates to the maze navigation module is the Locomotion module. In this module, there are 6 objectives that, once completed, will signify a completed locomotion module. The 6 objectives are: to be able to drive in any direction, to rotate in place, to detect black line intersections, to follow the black line optimally, to maneuver over speed bumps optimally, and to detect walls using distance sensors. Currently, our robot can move using Mecanum wheels in specified directions (forward, reverse, strafe left, strafe right) and can also rotate in place using a Bluetooth serial dongle combined with a command interpreter. This is only for debugging purposes as no wireless communication is allowed during the competition. This module will continue to be worked on into the next semester during which, line following and intersection detection will be implemented using 3 or 6 array line sensors.

The last module is the Image Processing module. In this module, there are two key groups with two objectives each: camera motion which includes panning the camera 90° both clockwise and counter-clockwise and also tilting the camera 5° up and down, and character recognition which includes detecting characters on walls successfully from 6 inches away and also storing the recognized character onto a USB flash drive. Currently, several characters can be recognized at a time from 6 inches away. This is irrelevant of a slight skew of text and variations in font. The servo for panning the camera has been tested to work using a function generator, but has not yet been implemented. The coming semester, work will continue within this module which will include implementing the camera pan servo and also quality and consistency testing for recognition of text characters.

1. **Engineering Specifications and Constraints**

The competition has specifications which involve what teams can use to build their robot and sizing of the robot and the maze board. The specifications of the maze board can be see below in Figure 4.

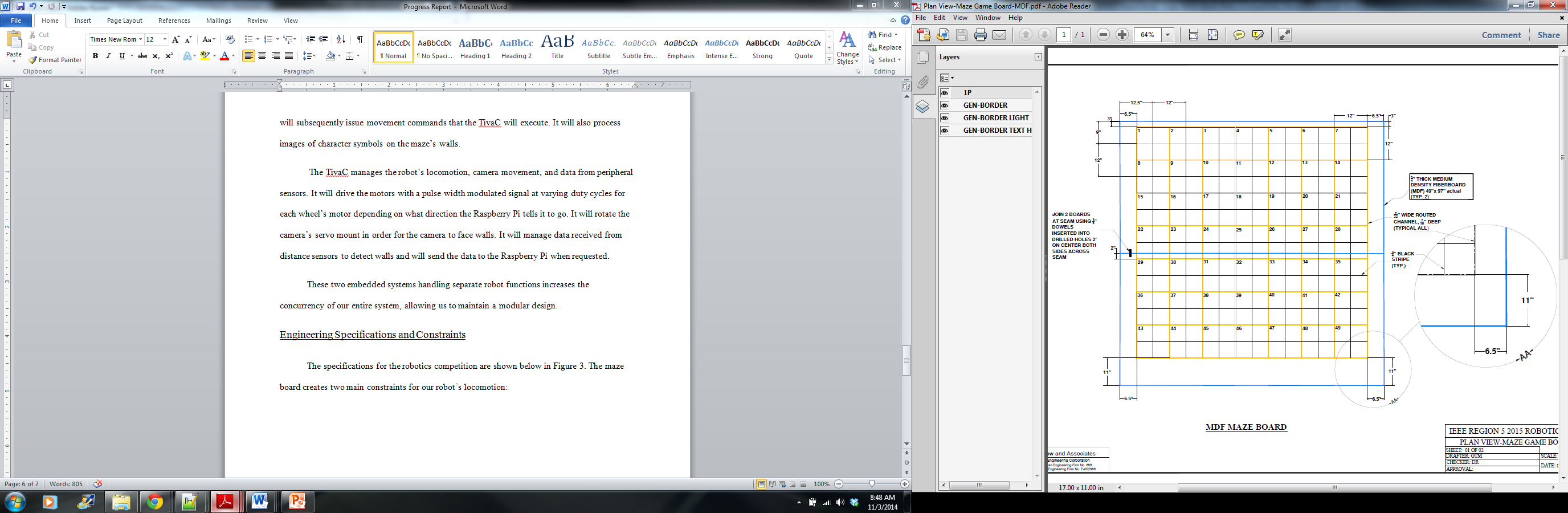


Figure : Maze board specifications for the IEEE R5 2015 Robotics Competition [10]

Each square cell of the maze is a fraction less than one square foot. Because of this, our robot, to be successful is suggested to be about a 7 inch square. This limits the sizing of many of our components including our chassis and motors. To test these conditions, a Computer-Aided Design (CAD) was used to estimate dimensions of our robot and how it would fit into each square cell of the maze. These estimates can be seen below in Figure 4a and 4b.

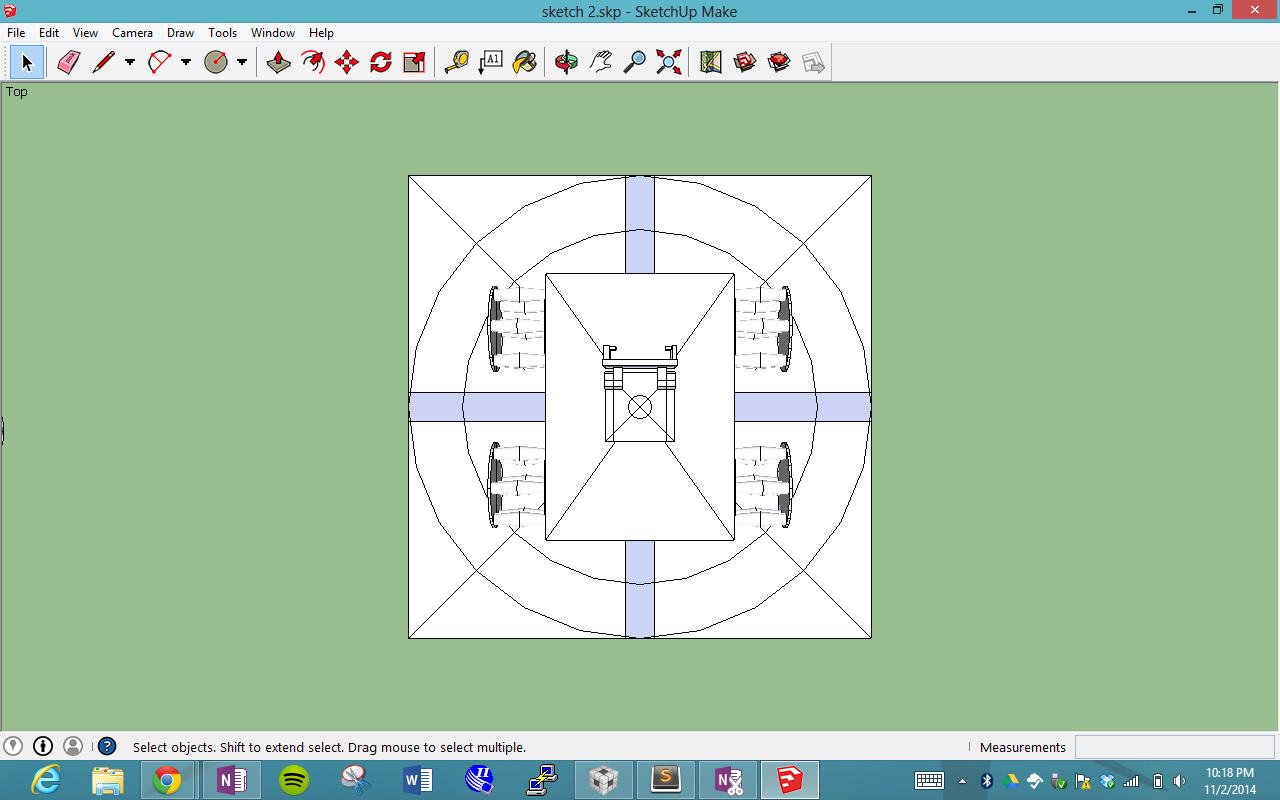
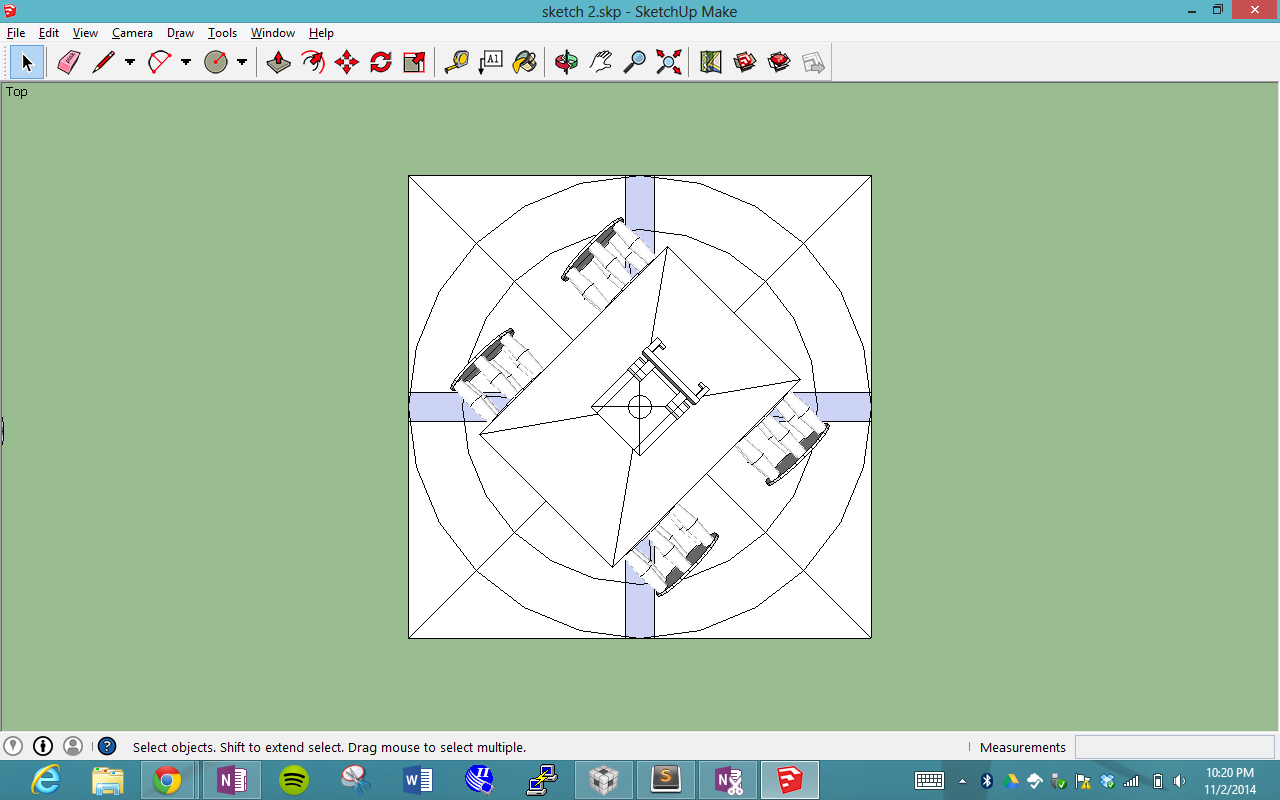
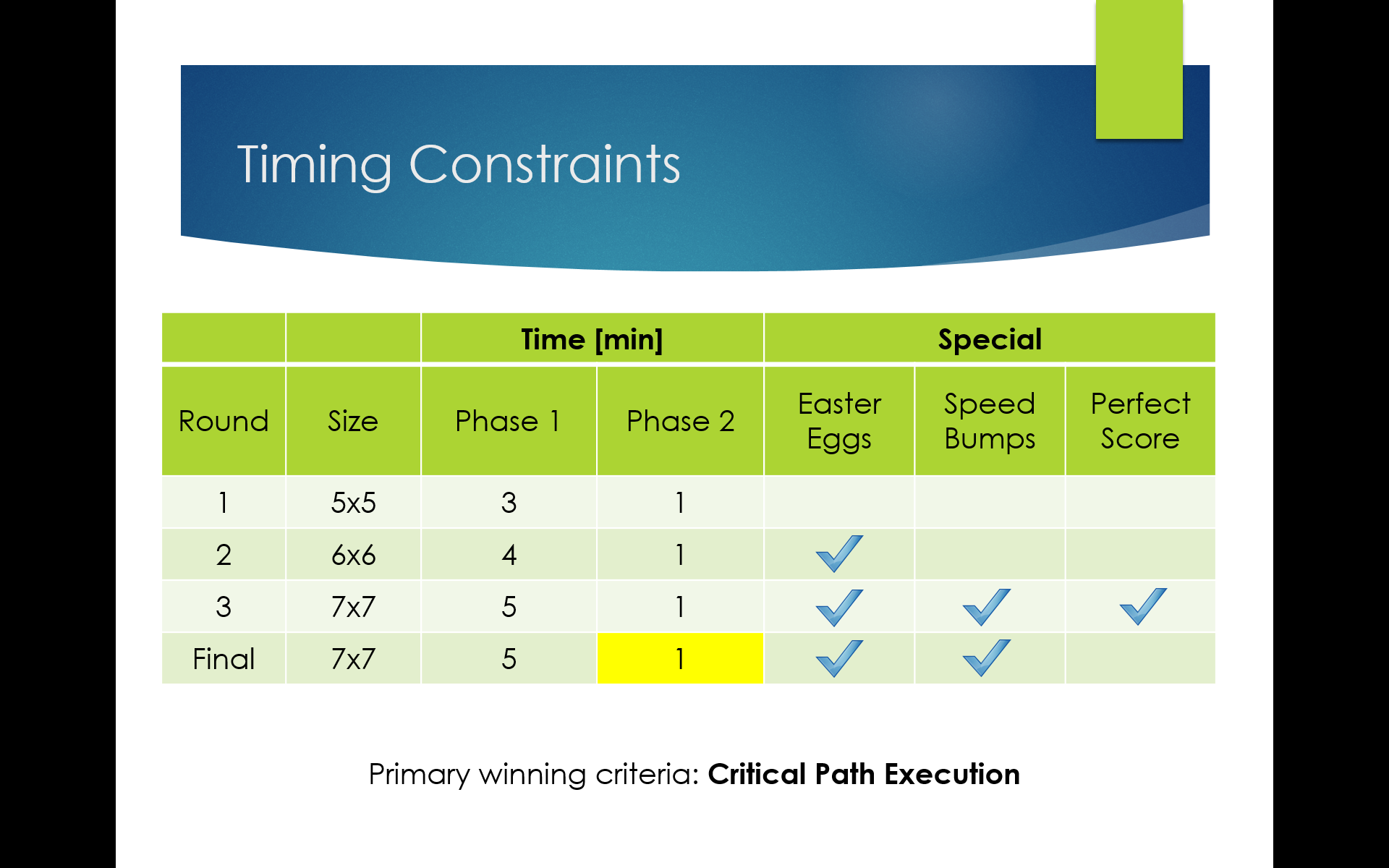
 

Figure a: 7' x 7' robot in a 12' x 12' square Figure 4b: The robot rotated 45 degrees at central pivot

Using the CAD as a reference, we were able to confirm that our design would fit well within the square foot maze cell with about an inch of clearance on either side of the robot at its widest rotation.

Timing specifications are also defined for the competition. Each round is listed below in Table 1, with each round detailed by time and special events.

Table 1: Timing Specifications per Round



Each of these timing specifications are important as once special cases are involved, such as speedbumps and Easter eggs, those only at more time to our overall runtimes. This can be a problem is we cannot complete the maze within allotted times.

We have a budget constraint and also a time constraint. As Dr. Glover is funding our robotics project, an excess of money cannot be spent. Each component has to have a distinct advantage confirmed with research and they have to be able to be reused in future projects of the department. As student, we also have a time constraint. As all of the members of our group are full-time students, the amount of time available is less than that of a full-time employee.

1. **Budget**

The total projected budget, split into cost of labor and cost of inventory, can be seen in Tables 2 and 3 below.

Table 2: Total projected labor budget

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Consultant Name** | **Task** | **Cost per Hour ($)** | **Hours per Week** | **Number of Weeks** | **Task Total** | **Consultant Cost** |
|  |  |  |  |  |  |  |
| **Ernie Patenia** |  | 75.00 |  |  |  |  |
|  | State Machine |  | 10 | 4 | 3,000.00 | $15,000.00 |
|  | Character Recognition |  | 10 | 4 | 3,000.00 |
|  | Misc. Task |  | 20 | 6 | 9,000.00 |
|  |  |  |  |  |  |  |
| **Victor Brzeski** |  | 75.00 |  |  |  |  |
|  | Locomotion |  | 15 | 4 | 4,500.00 | $15,000.00 |
|  | Camera Motion |  | 10 | 2 | 1,500.00 |
|  | Misc. Task |  | 20 | 6 | 9,000.00 |
|  |  |  |  |  |  |  |
| **Samuel Stephen** |  | 75.00 |  |  |  |  |
|  | Optical Analysis |  | 15 | 4 | 4,500.00 | $15,000.00 |
|  | Camera Motion |  | 10 | 2 | 1,500.00 |
|  | Misc. Task |  | 20 | 6 | 9,000.00 |
|  |  |  |  |  |  |  |
| **Fergyl Estioco** |  | 75.00 |  |  |  |  |
|  | State Machine |  | 10 | 4 | 3,000.00 | $15,000.00 |
|  | Locomotion |  | 10 | 4 | 3,000.00 |
|  | Misc. Task |  | 20 | 6 | 9,000.00 |
|  |  |  |  |  |  |  |
| **Dr. John Glover**  **(Facilitator)** |  | 150.00 | 4 | 8 | 4,800.00 | $4,800.00 |
|  |  |  |  |  |  |  |
|  |  |  |  | **Total Labor Cost:** | | **$64,800.00** |

Table 3: Total projected purchase costs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Supplier** | **Cost per Unit ($)** | **Quantity** | **Total Cost** |
|  |  |  |  |  |
| **Robot Body:** |  |  |  |  |
| Motors | Pololu | 19.95 | 4 | $79.80 |
| Motor Mounts | Pololu | 7.45 | 4 | $29.80 |
| Mounting Hubs (2-Pack) | Pololu | 6.95 | 2 | $13.90 |
| MecanumWheels (4-Pack) | SparkFun | 74.95 | 1 | $74.95 |
| Acrylic Sheet (12"x12") | Amazon | 5.25 | 3 | $15.75 |
|  |  |  |  |  |
| **Camera/Image Processing:** |  |  |  |  |
| Raspberry Pi B+ | Raspberry Pi | 39.15 | 1 | $39.15 |
| Raspberry Pi Camera Module | Amazon | 26.91 | 1 | $26.91 |
| Ultrasonic Distance Sensor | Amazon | 8.53 | 3 | $25.59 |
|  |  |  |  |  |
| **Camera Motion:** |  |  |  |  |
| 2-Axis Tilt Camera Mount | Amazon | 7.99 | 1 | $7.99 |
| Servos | Amazon | 7.66 | 2 | $15.32 |
|  |  |  |  |  |
| **Maze Supplies:** |  |  |  |  |
| MDF Wood Sheets (4'x8') | Home Depot | 59.96 | 1 | $59.96 |
| Fluted Dowels | Home Depot | 2.89 | 1 | $2.89 |
| Paint (Various Colors) | HomeDepot/Lowe | 14.38 | 5 | $71.90 |
| Misc. Painting Supplies | Walmart | 24.63 | 1 | $24.63 |
|  |  |  |  |  |
|  |  | **Inventory Total:** | | **$488.54** |
|  |  | **Total Labor Cost:** | | **$64,800.00** |
|  |  | **Total Cost of Project:** | | **$65,288.54** |

Expenditures to date and projected expenditures to date are shown in Tables 4 and 5 below.

Table 4: Total expenditures to date

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Supplier** | **Cost per Unit ($)** | **Quantity** | **Total Cost** |
|  |  |  |  |  |
| **Robot Body:** |  |  |  |  |
| Motors | Pololu | 19.95 | 4 | $79.80 |
| Motor Mounts | Pololu | 7.45 | 4 | $29.80 |
| Mounting Hubs (2-Pack) | Pololu | 6.95 | 2 | $13.90 |
| MecanumWheels (4-Pack) | SparkFun | 74.95 | 1 | $74.95 |
|  |  |  |  |  |
| **Camera/Image Processing:** |  |  |  |  |
| Raspberry Pi B+ | Raspberry Pi | 39.15 | 1 | $39.15 |
| Raspberry Pi Camera Module | Amazon | 26.91 | 1 | $26.91 |
| Ultrasonic Distance Sensor | Amazon | 8.53 | 3 | $25.59 |
|  |  |  |  |  |
| **Camera Motion:** |  |  |  |  |
| 2-Axis Tilt Camera Mount | Amazon | 7.99 | 1 | $7.99 |
| Servos | Amazon | 7.66 | 2 | $15.32 |
|  |  |  |  |  |
| **Maze Supplies:** |  |  |  |  |
| MDF Wood Sheets (4'x8') | Home Depot | 59.96 | 1 | $59.96 |
| Fluted Dowels | Home Depot | 2.89 | 1 | $2.89 |
| Paint (Various Colors) | HomeDepot/Lowe | 14.38 | 5 | $71.90 |
| Misc. Painting Supplies | Walmart | 24.63 | 1 | $24.63 |
|  |  |  |  |  |
|  |  | **Inventory Total:** | | **$364.52** |

1. **Conclusions**

This semester, great progress has been made on our robot as each module of the design works independently. In the coming months, significant progress is expected, including but not limited to drafting more optimal designs for the robot chassis, using line following for maze navigation, optimizing the maze navigation algorithm to account for certain case issues, and optimizing the image processing for reduced CPU overhead . During this time, each module will be tested rigorously to ensure that the final product has the best solution in each of the modules separately and the robot as a whole. If a module is found to lack robustness, then a replacement module will be researched, designed, and implemented. At this rate, being able to complete the maze and score at least 85% of the possible points in the competition are much more attainable.

1. **Works Cited**

[1] Image of IEEE Region 5 2015 Robotics Competition Track - <http://r5conferences.org/robotics->competition/

[2]Image of Raspberry Pi MicroController- https://lmshop.s3.amazonaws.com/item\_photos\_original/Raspberry\_Pi.jpg

[3]Image of Maze Solving Algorithm - http://www.cbc.ca/news2/pointofview/maze.jpg

[4]Image of Raspberry Pi Camera Module - <http://cpc.farnell.com/images/en_CC/rasp-picamera> -img-vert.png

[5] Image of Ultrasonic Distance Sensor - <http://1.bp.blogspot.com/-> dc78WX5EB0Y/UgnTePvjBrI/AAAAAAAAAtA/I9XItMAPYww/s1600/parallax-ping ultrasonic-sensor-large.jpg

[6] Image of TivaCMicroController - www.ti.com/ww/en/launchpad/img/launchpad-tivac- 01.jpg

[7] Image of Mecanum Wheels - https://cdn.sparkfun.com//assets/parts/7/5/6/7/11578-02.jpg

[8]Image of Motors - http://www.robotshop.com/media/files/images/pololu-12v-100-1-gear- motor-encoder-a.jpg

[9] Image of Servo Mount - http://www.foxtechfpv.com/images/7288\_3.jpg

[10]Image of IEEE Region 5 2015 Robotics Competition Track Specifications - <http://r5conferences.org/robotics->competition/

[11] Definition of Back Electromotive Voltage - http://en.wikipedia.org/wiki/Counter- electromotive\_force

1. **Appendix**

Table A1: squareNode class diagram

|  |  |  |
| --- | --- | --- |
| CLASS: squareNode | | |
| Data Type | **Member Variable Name** | **Description** |
| int | number | Number of the node in relation to the entire map. |
| bool | wall[4] | Index values 0,1,2,3 correspond to the North, East, South, and West faces of a square. For example, if wall[0] is true, that means a wall exists on the north face of this squareNode. |
| char | asciiSymbols[4] | Index values 0,1,2,3 correspond to the North, East, South, and West faces of a square. For example, if asciiSymbols[3] is true, that means an ASCII character exists on the south face of this squareNode. |
| bool | finished | If this is true, we do not need to perform a check for the existence of walls or ASCII characters for this squareNode. |
| stack<int> | unexploredPaths | This is a stack that indicates whether or not there are unexplored paths that branch out from this squareNode. |
| \*squareNode | adjacentCells[4] | This is an array of the squareNode addresses adjacent to this squareNode. Index values 0,1,2,3 correspond to the North, East, South, and West faces of the node. If the robot were to travel South from this current squareNode, it would ultimately end up in the squareNode whose address is located in adjacentCells[3]. |

Table A2: virtualMap class diagram

|  |  |  |
| --- | --- | --- |
| CLASS: virtualMap | | |
| Data Type | **Member Variable Name** | **Description** |
| \*squareNode | Start | This is the address of the starting squareNode of the maze. Before the competition begins, this startingsquareNode is known. |
| \*squareNode | Finish | This is the address of the ending squareNode of the maze. Before the competition begins, this endingsquareNode is known. |
| \*squareNode | currentNode | This is the address of the current squareNode that the robot is in. |
| int | currentNodeNumber | This is the number of the currentSquareNode that the robot is in. |
| int | mapSize | This is the size of the map in squareNodes. Before the competition begins, this size is known. |
| int | squaresVisited | This is the number of unique squareNodes that the robot has visited. The robot must travel a number of squareNodes equal to the mapSize in order to have a successful run. |

Table A3: navigationSystem class diagram

|  |  |  |
| --- | --- | --- |
| CLASS: navigationSystem | | |
| Data Type | **Member Variable Name** | **Description** |
| virtualMap | Map | This is the virtualMap of the playing field. |
| int | currentFace | This is the cardinal direction that the robot is currently facing. 0,1,2,3 corresponds to North, East, South, and West. |
| int | currentSquare | This is the number of the squareNode that the robot currently resides in. |
| stack<int> | criticalPath | This is a stack of node numbers that make up the critical path. It is updated as the robot travels along the maze. |
| stack<int> | travelHistory | This is a stack of node numbers that the robot has most recently visited. It used for retracing steps. |
| stack<int> | revisitPath | This is a stack of node numbers that the robot must revisit because there are still unexplored paths that branch out from those nodes. |
| queue<int> | asciiLocations | This is a queue of node numbers that indicate that there is at least one ASCII character in the walls of that node. |