**Summer** 14

The AWESOM-O: Competitive Robot Team

**Ryan Simms | Ryan Penny | Mark Power | Joshua Odagwe |Afam Edozien**

**200754166 | 201140027 | 201019643 | 201242864 | 201104221**

**E N G I 1 0 4 0: M e c h a n i s m s a n d E l e c t r i c C i r c u i t**

**Table of Contents**

**1. INTRODUCTION** **3**

**2. STRATEGY** **3**

**3. DESIGN** **4**

**4. PROJECT MANAGEMENT** **8**

**5. DISCUSSIONS AND RECOMMENDATIONS** ..**9**

5.1 THE OVERALL SUCCESS OF THE TEAM ..9

5.2 THE COMPETITION 10

**6. APPENDIX A: DESCRIPTION OF THE ALGORITHMS** ………………………………….....…**12**

**1. INTRODUCTION**

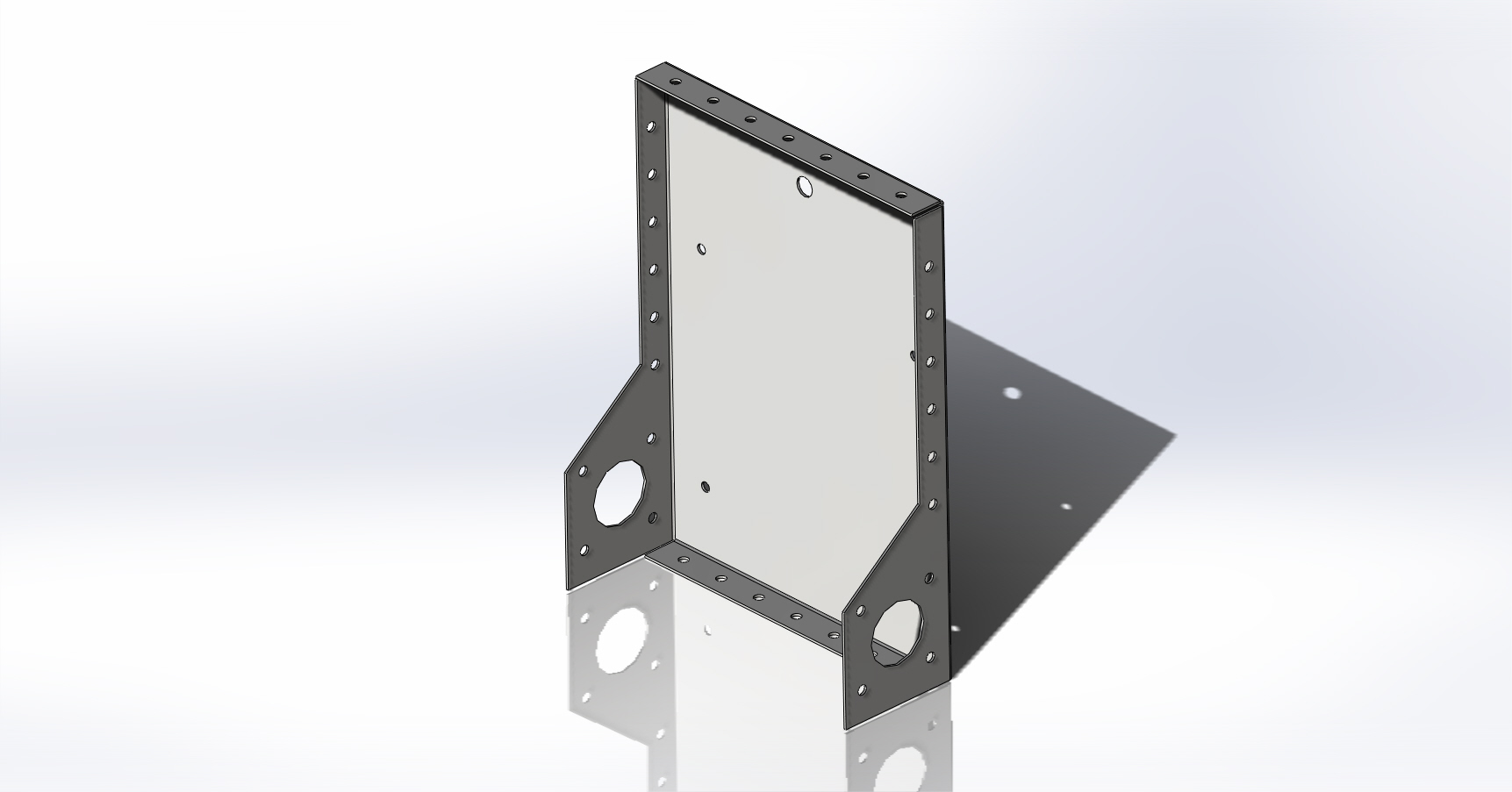
Throughout the past three weeks, the AWESOM-O Robot Team has designed and created a robot to compete in the Engineering 1040 Robot Competition. Each Team in the competition was given an additional IR sensor, stepper motor, and a solenoid and was also allowed to obtain a square foot sheet of aluminum from the machine shop to aid in building the transporting device. The competition required each team to build a robot that carries, drags, launches or pushes spindles worth varying points into a collection area designated to each team. Spindles located on the opposing team’s territory are considered “high value” and are worth double the usual number of points but disqualification occurs if the encroaching robot is tagged by the “home team” robot on the contender’s side. Each team competed in 3 preliminary matches with their robot, and the top eight teams continued to a knockout round. However the top eight teams were identified based on the total score in the preliminary matches. The top eight teams competed in a knockout stage and then four teams proceeded to the semifinals. The two winners proceeded to the finals to determine the overall class champion as well as the runner up. The aspects of the robot developed by the team included a mechanism for pushing the spindles into the collection area, and an algorithm for navigating the course.

**2. STRATEGY**

In order to perform at the highest level possible during the robot competition on August 8th, 2014, it was essential to develop a strong competitive strategy. The robot designed and created by the AWESOM-O consists of three IR sensors that were used to aid in directing the robot to the safest and most promising path to acquire the highest number of points. The sensors allow the robot to follow the left and right wall, turn at specific points and distance, recalibrate its position, collect spindles and push them to the collection area. It was determined by the team after a lot of brainstorming that making the robot relatively small in size and utilizing the sensors was the best idea. The AWESOM-O agreed after much discussion that the best method was to keep the robot on its own side, relying on the sensors for the majority of the competition, as being eliminated from the match was a risk not worth the point contribution from getting tagged by the opponent’s robot. However the team’s strategy is a defensive one; the robot will score the maximum amount of points in the safest manor. The team however agreed to use two codes for the competition.

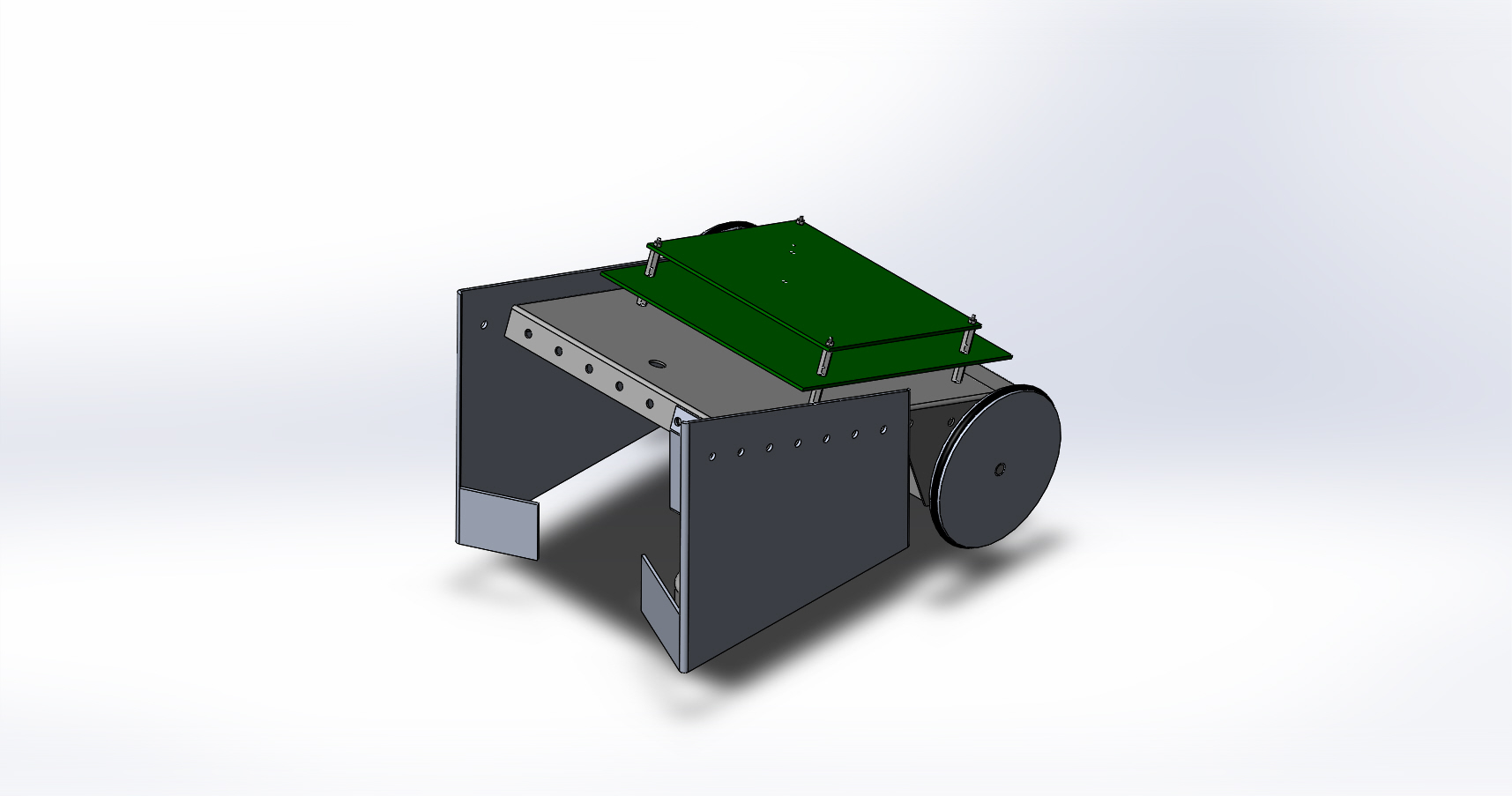
**3. DESIGN**

The AWESOM-O group members worked as a team to create the final design. Various tasks were given out to every member of the group. The robot itself was to be inclined at a very small angle of 30 degrees. The frame of the robot without the wheels and the stepper motors can be seen in Figure 1.



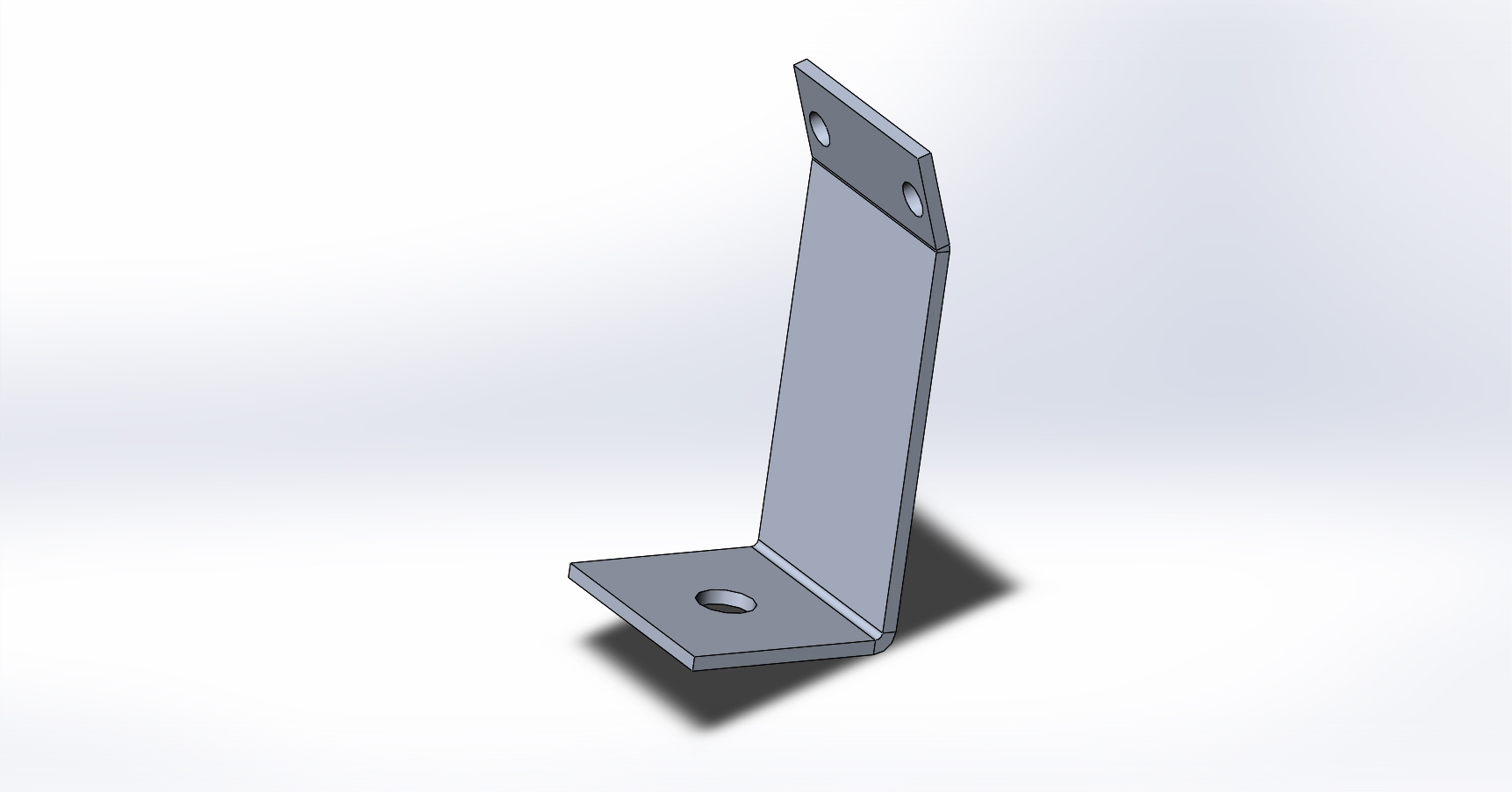
**Figure 1: Basic Frame of the Robot**

The final robot design consists of a spindle collecting plow, a wheel bracket to hold the ball, and three IR sensors. Two of the IR sensors are located on the sides of the spindle collecting plow. The design proved to be simple, effective, fairly cost efficient and allowed for easy use and programming. The final design of the robot without the IR sensors can be seen in Figure 2.



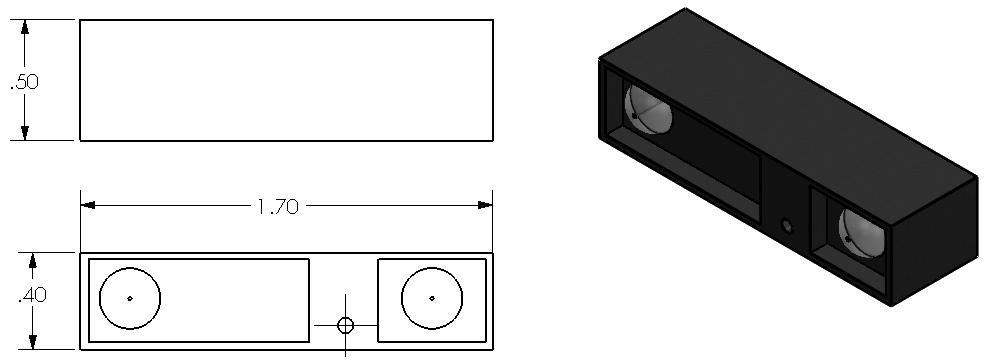
**Figure 2**.

In other to hold the collected spindles from going out easily, the team agreed that the edge of the plow be bent in at about 45 degrees to allow the spindles slide in. An IR sensor was put on both sides of the plow to make robot follow walls. The plow is made out of Sheet metal and has seven holes on each side. Four of the holes were being attached to the robot with the aid of screws. A wheel bracket was created on the front of the robot and is made out of Sheet metal. The wheel bracket has two holes at the top and one hole at the bottom and it’s attached to the robot with the aid of screws. However, the top part of the bracket is inclined at the same angle as the robot while the bottom part of the bracket is attached to the ball with a screw. The robot is designed in such a way that the spindles slide into the open area in the plow. When all the spindles are collected, the robot pushes all collected spindles to the collection area. The final design of the wheel bracket can be seen in Figure 3.



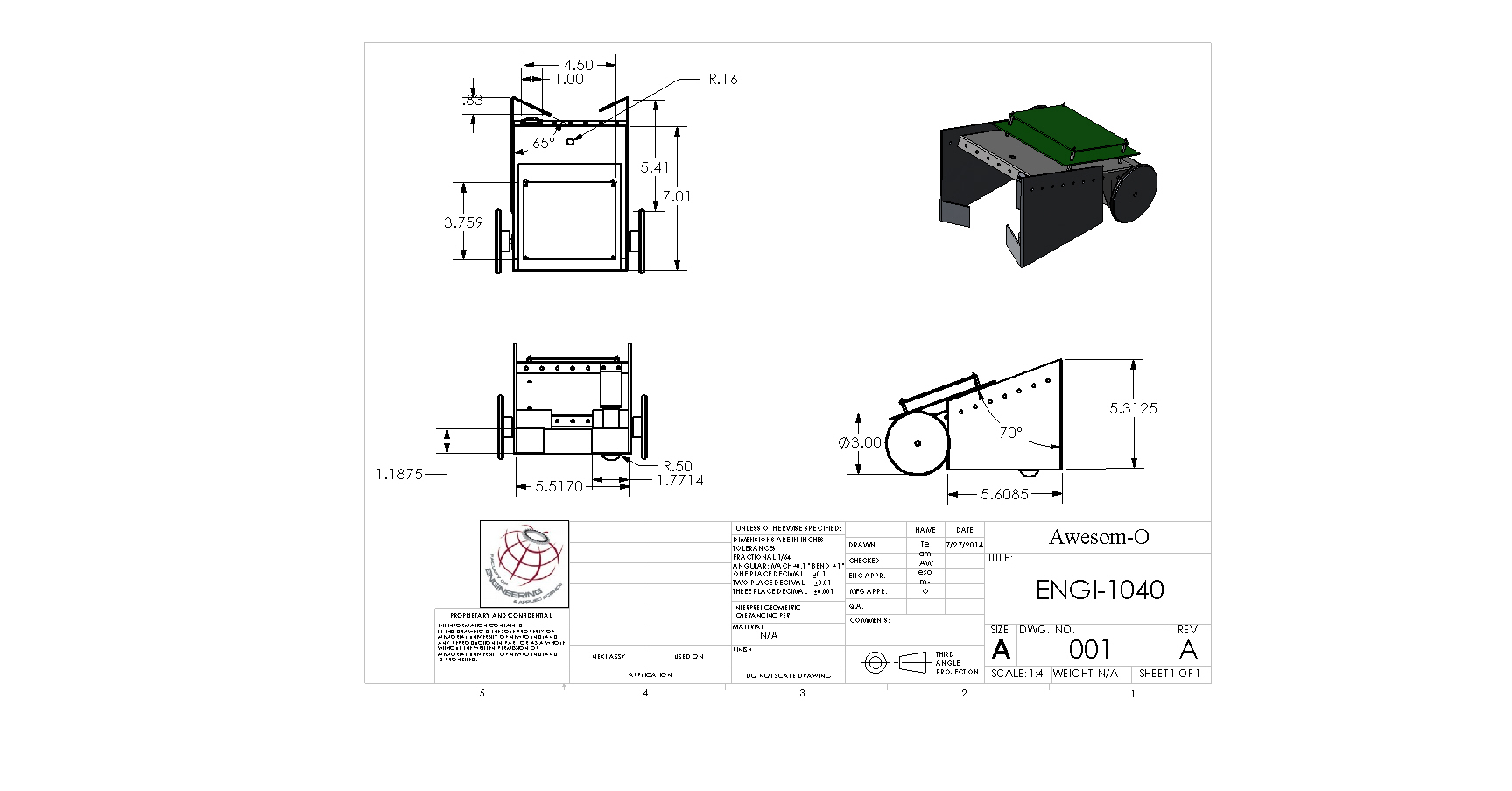
**Figure 3**.

In order to control the movement of the robot, three IR sensors were used. One sensor was attached to the left side of the robot which allows it to follow the left wall, and another was attached to the right side in order for the robot to follow the right wall. There is one more sensor located in front of the robot. The front sensor senses obstacles and walls in the course. These sensors told the robot when and where to turn, collect the spindles, and allowed it to recalibrate its position along a wall. A representation of the sensor is shown in Figure 4.



**Figure 4: 3D representation the sensors used on the robot**

The AWESOM-O team worked diligently and divided the tasks amongst everyone. The drawing of the design can be seen in Figure 5.



**Figure 5.**

The overall cost of the robot was kept to a minimum, with the most expensive material being the sheet metal. There were many pieces of the robot that were obtained from the mechatronics lab, some of which include the sensors, wires, fasteners. All of the materials given to the team in the lab were borrowed and returned upon the completion of the robot competition. The AWESOM-O obtained a square foot sheet of aluminum from the machine shop to aid in building the transporting device.

**Table 1 – Cost Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Material** | **Quantity** | **Unit Price** | **Total Price** |
| Sheet Metal | 4’ x 8’ sheet | $100/ 4’ x 8’ sheet | 100.00 |
| MunderBoard\*\* | 1 | $55 | 55.00 |
| **TOTAL** |  |  | **$155.00** |

\*The price for Sheet Metal was found on the course website instruction (9) at: http://www.engr.mun.ca/~migara/eng1040/\*\*

**4. PROJECT MANAGEMENT**

A crucial component to the success of the competition was the management of the project and the team. While planning for the entire design, manufacturing and testing process, some key factors were kept in mind, such as the time constraints and the competence and skill set of each team member. Since there were just three weeks to develop a winning robot, the team had to utilize each member’s strong suits to their fullest potential. For maximum efficiency, continuous project phases were identified and deadlines were created for each phase. The first phase was designated to the design of the robot. The team proceeded in brainstorming and each member created and communicated an idea that they thought would best allow the robot to move around the track as well as push spindles in the collection area. Once the group decided on a final design, phase two began. During this time the initial Solid works representation of the robot was created, as seen in Figure 5, and the plow for the robot which is made of sheet metal was manufactured in the machine shop. The next phase was to completely assemble of the robot; this included properly wiring the robot, putting together each component including all motors and the three IR sensors. Phase four called for the perfection of the code that controlled the robot. The entire group worked strongly to create logical algorithms and aided in solving issues with the code. The final phase was testing the robot. This phase caused for plenty of trouble shooting and perfecting issues involving technical elements, design, and structure of the robot. Throughout each phase of the project, each member remained highly involved to ensure that everybody maintained knowledgeable on how the robot worked, as well as how and why any part of the robot was changed or improved upon. The team met 3-4 times a week, and daily during the week of the competition to ensure that each member was on the same page and to determine any deliverables that have been and still needed to be met.

**5. DISCUSSIONS AND RECOMMENDATIONS**

**5.1 THE OVERALL SUCCESS OF THE TEAM**

During the design, manufacture, coding, and testing of the robot, the AWESOM-O successfully accomplished a working robot by means of a strong work ethic on an individual and group scale. Despite technical difficulties with communications, the team was able to communicate effectively. A Facebook message thread and email was used to keep code versions, deadlines and progress consistent across the team. Goals were agreed upon during meetings, and efforts were made to ensure every team member understood all aspects of the project, such as the code. Every team member worked on every aspect of the project to some extent, allowing us to make progress even if some team members were absent, and everyone to have independent ideas for improvements. If one team member was encountering difficulties with the spindle collection mechanism, for example, another could offer input or different ideas for improvement. One success of the team was the time spent in the planning stages, only fully assembling the robot when the design was finalized. The time spent in the planning stages was spent on fully designing the different aspects of the robot and how they would work together. This meant that less fine tuning was required to perfect the project. When issues with the project did arise, the team was able to effectively troubleshoot and fix the problems with the robot. One example was when the robot did not properly execute the follow wall function. To solve this problem, each component of the robot involved in the function was independently tested. It was found that the left sensor was not working, and there was also a logical error in the code. Solving both these problems allowed our robot to perform the function appropriately in the competition. Even though the team worked well together, improvements could have been made. If this project were to be completed again, adjustments would have been made as follows:

* More detailed timeline for each phase, as well as what each phase fully entailed. This would allow the team to stay on top of the project from the beginning, without leaving any major changes or alterations until the last minute.
* More research should have gone into the design phase. A lack of ‘thinking ahead’ led to a major alteration in the design.
* Roles would be more clearly defined in the group to ensure everybody knew their expectations. Each member of the group did contribute significantly to the success of the project, but time could have been used more efficiently if positions clarified and very specific tasks were given to each member.
* ‘Meeting Minutes’ would have been done to track all of the deliverables, as well as to be used for reverence. A lot of communication and scheduling was executed via group e-mails, but tasks were rarely specified in these e-mails.

**5.2 THE COMPETITION**

A strategy was devised for the day of the competition. The team found which groups they were competing against and when possible, the best path could be determined for each match based on a basic knowledge of the competitor’s path. Since the robot was calibrated to push the spindles into the collection area, the batteries where charged fully to ensure the robot would not run out of power during the competition. The batteries were also charged between matches to keep the power at an optimal level. By testing the robot, it was found that each round that the robot pursued without fully charged batteries would alter the fine tuning of the path, which cumulated to cause error in the movement of the robot.

For the first round of the competition, the team agreed to use the second code which included a delay function to stop at the middle of our zone for a given amount of time. However, the opposing team’s robot got stuck while entering their own box and could not finish the track. Our robot then went on to collect the spindles on its own side after the delay time was over. The team successfully got to the collection area with all the spindles in its zone and was declared the winner as we had more points than the other team.

For the second round of the competition, the team agreed to use the first code which included follow wall function. However, the robot got stuck in the box while attempting to take its own spindles on its side. This code did not work perfectly as the “followRightWall ()” function did not fully adjust the robot. The robot was not able to get any spindle to its collection area as it was stuck in the box, resulting in a loss to the opposing team.

For the third round of the competition, the team agreed had to choose between two codes to use for the final round. The AWESOM-O after much discussion agreed to use the same code which was used for the second round which included the follow wall functions. The robot performed as expected and went into the opposing team side to take one of its spindles. The robot successfully made a quick 180 degrees turn and followed the right wall function back. This round was a success as the robot got the highest number of points in this round. However, this round was declared a draw as the opposing team had executed the same route as we had chosen. Both teams were given equal number of points.

After the three rounds, the top eight teams with the highest total number of points went on to compete against each other. The AWESOM-O were ranked 8th with 400 points and moved on to the next round. This round was a knockout round whereby the team which loses gets knocked out from the competition with the winning team advancing to the semifinals.

For the knockout stage of the competition which included eight teams, the AWESOM-O agreed to use the same code which had been used earlier for the last two rounds. The robot started as expected and went into the opposing team side to take one of its spindles. The robot successfully made a quick 180 degrees turn and followed the right wall function back. The robot followed the track well and took the spindles to the collection area. However, the robot malfunctioned and took out all of its spindles from its collection area to the middle area due to improper alignment. The round was highly considered a failure as the team ended up with no spindles in its collection area and was declared the loser, resulting in eviction from the competition. The opposing team went on to the semifinals of the competition.

If the project was completed again, some key points to reconsider would have been:

* Using the sensors was useful for tolerances and detecting obstacles, however the sensors also responded to advancing contenders. If an opposing robot approached the robot head on, the front sensor would read the opponent as an obstacle, which resulted in the robot turning too soon, setting it off course. It would have been better to have a mapping algorithm so the robot knew fully where it was and when to turn, or to use more of the sensors so that the robot would still continue on course if there was a robot in front of it.
* More consideration should have been given to the paths of other teams. Nearly all the teams used the same route in the competition. A code could have been written to counter all those who used the same route. The program could have been modified to quickly pass into the opposition zone from the side to the middle then back into our own zone, thereby getting four of the opposing team’s spindles.

**APPENDIX A: DESCRIPTION OF THE ALGORITHMS**

**void rotateCW(int motorNum)**

Overview – rotates the first (motorNum=1) or second motor (motorNum=2) one step in the clockwise direction.

Input - motorNum – the number of the motor to be rotated. Only rotates first or second motor.

**void rotateCCW(int motorNum)**

Overview – rotates the first (motorNum=1) or second motor (motorNum=2) one step in the counterclockwise direction.

Input - motorNum – the number of the motor to be rotated. Only rotates first or second motor.

**int readFrontSensor(void)**

Overview – reads the sensor value for the front sensor.

Output – the front sensor value, an integer.

**int readLeftSensor(void)**

Overview – reads the sensor value for the left sensor.

Output – the left sensor value, an integer.

**int readRightSensor(void)**

Overview – reads the sensor value for the right sensor.

Output – the right sensor value, an integer.

**void goForward(void)**

Overview – moves the robot forward one step.

**void goForwardx(int stepNum)**

Overview – moves the robot forward a given number of steps.

Input – the number of steps to be moved forward.

**void goForwardF(void)**

Overview – moves the robot forward until the front sensor is sensed

**void goForwardR(void)**

Overview – moves the robot forward until the right sensor is sensed

**void goForwardL(void)**

Overview – moves the robot forward until the left sensor is sensed

**void turnRight(int stepNum)**

Overview – rotates the robot to the right a given number of steps.

Input – the number of steps to be rotated to the right.

**void turnLeft(int stepNum)**

Overview – rotates the robot to the left a given number of steps.

Input – the number of steps to be rotated to the left.

**void followLeftWall(void)**

Overview – follows the left wall until the front sensor reads a certain value

Input – sensor-- the front sensor value required to exit the function.

tol-- the minimum left sensor value for which the robot would move forward in a straight line.

**void followRightWall(void)**

Overview – follows the right wall until the front sensor reads a certain value

Input – sensor-- the front sensor value required to exit the function.

tol-- the minimum right sensor value for which the robot would move forward in a straight line.

**void followRightWallIt(int dist, int tol)**

Overview – follows the wall on the right a given distance

Input – dist-- the distance, or number of times the robot executes the loop.

tol: the minimum right sensor value for which the robot would move forward in a straight line.

**void followLeftWallIt(int dist, int tol)**

Overview – follows the wall on the left a given distance

Input – dist-- the distance, or number of times the robot executes the loop.

tol-- the minimum left sensor value for which the robot would move forward in a straight line.

**void followRightWallF(int sensor, int tol)**

Overview – follows the right wall until the front sensor reads a certain value

Input – sensor-- the front sensor value required to exit the function.

tol-- the minimum right sensor value for which the robot would move forward in a straight line.

**void followLeftWallF(int sensor, int tol)**

Overview – follows the left wall until the front sensor reads a certain value

Input – sensor-- the front sensor value required to exit the function.

tol-- the minimum right sensor value for which the robot would move forward in a straight line.