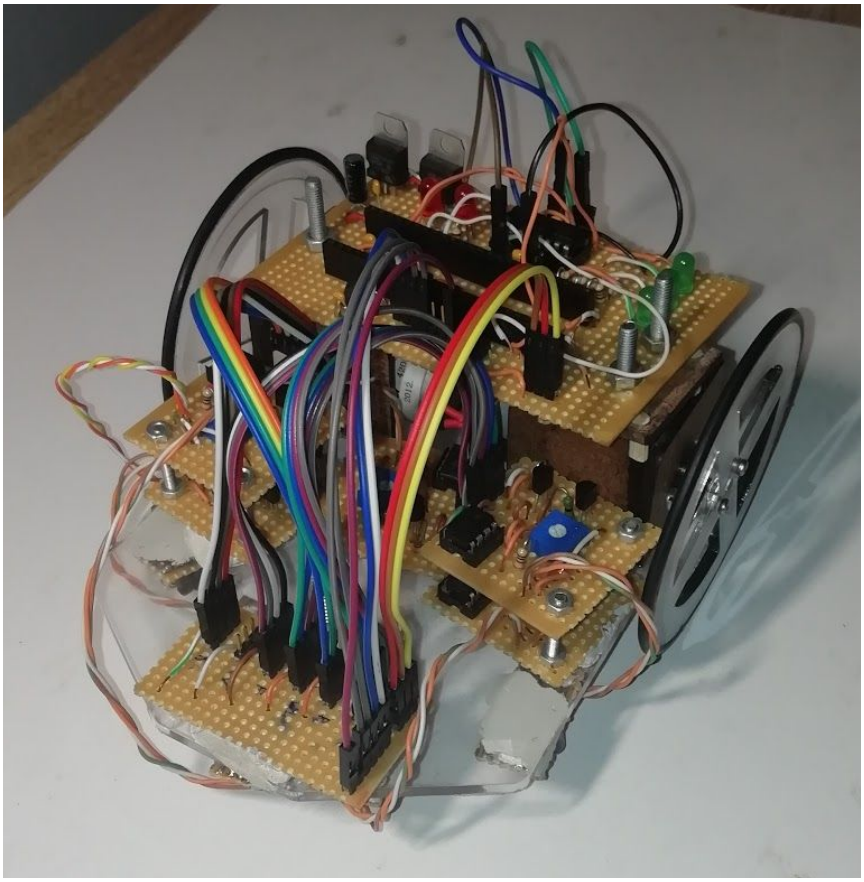
EEE3099S Milestone 3:

03/10/2019



Group Number	15
Bright, Daryn	BRGDAR004
Friedman, Jonathan	FRDJON009
Njoroge, Mark	NJRMAR003
Patrizi, Gianluca	PTRGIA002

0 Work Distribution

Table 1 Work distribution for Milestone 3

Person(s) of Responsibility	**Task Descriptions**
BRGDAR004	Sub-sytem Description, Testing Procedure and Testing Results, Commentary on Mechanical Design, Conclusion
FRDJON009	Algorithm and systems, Algorithm testing, system results, commentary on Algorithm
NJRMAR003	System Description, testing , Commentary on Electrical design, Conclusion, Circuit Schematic
PTRGIA002	Introduction (1.1-1.4), Electrical Sub-System Description, Testing and Results, Veroboard Schematic

Physical Testing of the robot was done by all members in White Lab.

1 Introduction

1.1 Introduction

This report discusses the processes undergone in developing a vehicular robot that follows, and solves, a line maze, in the quickest time possible, by explaining and examining the various stages of its development and criticizing key realisations throughout these stages.

1.2 Aim and Objective

The aim of this project is to create a robot that can; explore a line maze in its entirety, leaving no pathway unexplored, and then calculate and travel along the shortest possible path from the starting point to the finish, in the shortest time possible. The end of the maze is indicated by a black circle with a 150mm diameter which the robot must stop on to prove that it has found the end of the maze.

1.3 Design User Requirements

There were several requirements set out by the client that the robot design needed to satisfy in order to be accepted; they are as follows:

Whilst line sensing the sensor circuitry must output a HIGH for line detection and a LOW otherwise. The robot must be supplied by a 2 cell LiPo battery. The robot has a size restriction of 150 mm x 150 mm x 100 mm (width, length, height). The robot must use an STM32F0 family microcontroller. The robot must start at the push of a pushbutton and sense the end of the maze marked by a 150mm diameter black circle. Once the robot has explored the entire maze it must stop and once it has calculated the shortest path through the maze, blink an LED. The robot, once replaced at the start of the maze, must travel along the shortest path at the press of a push button. The robot design must use the two motors and the sheet of Perspex supplied by the client.

1.4 Design Technical Requirements

The robot will use an STM32F051C6 microcontroller to do all its processing and maze solving. The robot will use voltage regulators to output controlled voltages from the LiPo battery. It will use Infrared LEDs and phototransistors to detect the black line. The chassis of the robot will be cut from the supplied Perspex according to the size requirements. The circuitry will include an easily accessible pushbutton used to start the robot initially and when it will travel along the shortest path. An LED will be used to indicate that the robot has explored and solved the entire maze.

2 Sub-system Description

2.1 Introduction

The robot will be made up of three crucial sub-systems, namely; the mechanical sub-system which entails the chassis design, wheel placement, and motor gearing ratios. The Electrical sub-system which includes the sensor circuitry, voltage regulation circuitry, and the STM320F51C6 circuitry. The algorithmic sub-system is what the robot will use to map and solve the maze and is implemented in the coding of the microcontroller.

2.2 Mechanical

The mechanical sub-system includes two motors (JSX 190-27 (1:190)) that were altered from their bought state. Their gear ratios were adjusted, by removing a gear train set with a gear ratio of 1:2.5. This changed their gear ratios from 1:190 to 1:76. Attached to each wheel was an 80mm diameter wheel laser-cut from perspex with a 16mm diameter hub, 6mm wide. This assembly attached to the single-layer chassis via a motor mount structure. Situated on the motor mount was a Veroboard mount on which the STM32F0 microchip, the motor drive, the voltage regulators and other components sat. The front wheel is a ball-and-socket type wheel that could rotate in any direction easily. Five smaller mounts were used to hold the sensors' sub-circuitries in place. This can all be seen in the following figure 1.

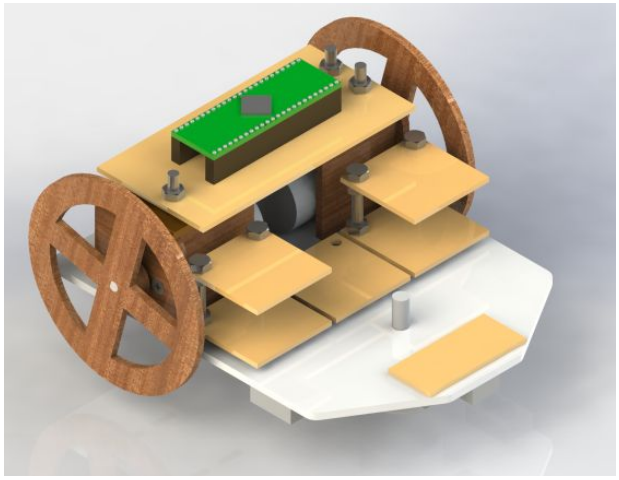


Figure 1: Rendered image of the final robot design

2.3 Electrical

The electricals of the robot were implemented in modularly to allow for easy debugging of the robot. The robot had five line sensors attached to its underside that were connected to their respective sub-circuits. The sensors' sub-circuits were then connected to a central board which provided them with 6V, 3.3V, GND from the STM circuit, and an Output to the STM circuit for each sensor.

The STM circuit included the voltage regulators, the motor driver, the indicator LED, the pushbutton and the STM32F051C6 chip. This board supplied all other sub-circuits/components with power and received all the outputs from the sensors. It used these sensor outputs as inputs to direct the robot around the maze. The 3.3V regulator was required for the STM board as that is its maximum safe operating input voltage.

The sensors used IR LEDs and Phototransistors because IR can be used to easily differentiate between black and white as black coloured objects absorb IR light and white coloured objects reflect it. The sensor circuits consisted of a comparator opamp that compares the input from the Phototransistor to a reference voltage set by a potentiometer. The more IR light that was reflected onto the phototransistor the higher the voltage to the positive terminal of the opamp would be and vice versa. The output of the comparator fed into a transistor-based circuit that would output TTL; 3.3V for HIGH when on black and <0.5V for LOW when on white.

Each of the 5 sensors had a very specific role. These roles are Left Sensing, Right Sensing, Forward Sensing, Left correcting and Right correcting. The sensor circuitry was based on circuitry found online [1]. Final circuit diagrams can be viewed in appendices 9.1 Figure 4 and 9.2 Figure 5.

2.4 Systemic (Algorithmic)

The STM32F0 microcontroller was used to control the robot. It received digital logic signals from the sensors and outputted logic signals to the motor driver IC. It also outputted Pulse Width Modulation signals to the enable pins of the motor driver to change the speed of the individual motors. The sensor data was used to correct the path of the robot and follow the line and to judge when to execute a turn. It was then used to stop the turn when the sensors were realigned with the track.

The Algorithm had two stages of requirements to meet. The first being to map out the maze autonomously and the second finding the shortest route from start to finish. It would use a coordinate system to map out each junction of the maze [2]. The robot could then navigate the maze learning all the points and choose turns based on where it had already been. Furthermore by keeping track of the number of possible turns at a point and the number of times that each point was visited it would be easy to ascertain when the maze had been explored in its entirety. At this point, the robot could navigate itself back to the end of the maze having learnt the whole maze with a high level of efficiency.
By keeping track of the sequence of points that are explored and having the coordinates of those points it is possible to construct an adjacency matrix of n X n terms where n is the number of points logged. This matrix will show the distance from one point to another where M[i][j] is equal to the distance from i to j. Using this matrix with Dijkstra's algorithm it is possible to find the route from beginning to end of the maze which has the shortest distance. [3]

3 Logical Testing of Subsystem and System Level Testing

3.1 Subsystem Testing Procedure

3.1.1 Mechanical

Each motor was tested by connecting it to a 6V power supply (4x1.5V AA batteries). This was to ensure the motor’s worked with a standard power supply. The motors had 100µF capacitors connected across their leads to protect them. As a result of the motor internals being altered, as previously mentioned, they were retested with the 6V power supply to ensure they still functioned properly.

3.1.2 Electrical

Each circuit/sub-circuit was first implemented and tested on a breadboard before it was soldered onto veroboard to ensure the circuit worked and that the output was strictly between 2.7-3.3V for HIGH and <0.5V for LOW.

Once each sensor was built it was individually tested. This was done by holding the sensor over a piece of black tape, holding it over a white piece of paper, moving it across from above white paper, over the black tape and back to above the white. These three tests were done at a height that was approximately where the sensors would be once attached to the robot.

Thereafter, all the sensors were placed on the underside of the robot and the robot was placed in various positions on the maze where certain sensors were expected to trigger and others not. A multimeter was used to measure if the sensors were triggered as expected.
This was done for all possible configurations on the maze as shown below in figure 2.

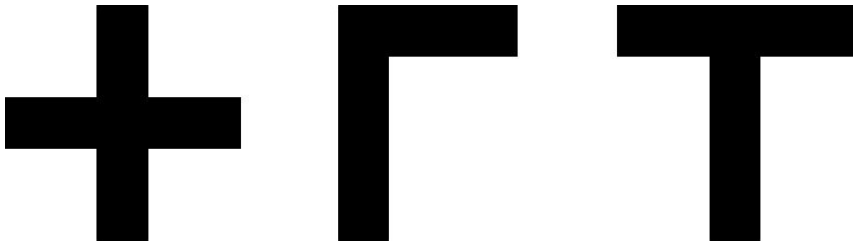


Figure 2: A crossroads, a single turn, and a T-junction, all with Dead-ends

3.1.3 Systemic (Algorithmic)

The algorithm was tested on a PC using keyboard inputs to simulate data received from sensors and printing turning decisions to the console. This allowed the algorithm to be tested before the robot was ready. The algorithm was tested on a number of mazes to find bugs and ensure that it would be robust on any map. Tests were also conducted on the STM to ensure that some of the longer processes could be run fast enough on the processor to be used while the robot would be in motion without disrupting its ability to stay on the line.

The motors were connected to the development board in order to test the effectiveness of pulse width modulation in speed control which was controlled by the STM board.

The final tests on the working robot involved testing its functions individually to ensure that the motors were controlled effectively and that the sensors were all functional. Tests were run to fine-tune the machine's orthogonal turning, its u-turns and its ability to follow a straight line. These tests were the most telling as they allowed us to see how the subsystems worked together and the additional challenges that would come from integrating them.

3.2 Subsystem Testing Results

3.2.1 Mechanical

In the initial testing of the motors with each one hooked up to a 6V power supply, the motors did not turn. After investigation, it was revealed that the problem was with the power supply as it was not connecting properly with the motors. After that problem was fixed, the motors ran smoothly, however, they had a low output rpm.

After the adjustment of the motors' internal gear trains, the motors, once again connected to the 6V power supply correctly, worked and turned at 2.5 times the speed that they had previously. This was satisfactory as the desired outcome was for the maze to be executed in a race fashion. This outcome decreased the output torque by the same ratio of 2.5 but was not an issue due to the already sufficient torque output of the motors.

3.2.2 Electrical

The primary breadboard testing showed that the sensor circuitry worked. The circuit would output HIGH (3.3V) when held over the black tape, LOW (<0.5V) when held over the white surface and would transition from LOW to HIGH to LOW when moved across from the white surface to the tape and back to the white surface. The response was noted to be quick.

Thereafter the sensors were built and retested in the same manner to ensure no soldering mistakes had been made. The sensors were then placed on the underside of the robot in suitable locations. The robot was placed on the maze at locations as shown in figure 2 above and the results can be seen in Table 2 below:

Table 2: Sensor Testing Procedure

Maze Configuration	**Expected Sensors to Trigger**	**Result**
Straight Line	Forward	Success
Straight Line but Skew	Forward and either Left correcting or Right correcting	Success
Right only	Right	Success
Left Only	Left	Success
Right and Forward	Forward and Right	Success
Left and Forward	Left and Forward	Success
4-Way Junction	Left, Forward, Right	Success
T-Junction	Left, Right	Success
Dead-end	None	Success

The testing of the sensors proved they had been implemented successfully.

3.2.3 Systemic (Algorithmic)

Tests run on PC showed that under ideal circumstances the algorithm worked on complex mazes with multiple loops. It successfully explored the mazes without making any wrong or redundant turns before navigating back to the ending point along the shortest path. It then successfully navigated the maze from start to finish along the path with the shortest distance.

PWM was effective as a speed controller once the frequency of the waveform was adjusted to one compatible with the switching frequency of the motor driver. This speed control was used for line tracking and the code was tweaked so that when the correcting sensors hit the black line the robot would slow the speed of the wheel on the side of the sensor for a short period of time.

Tests on the STM showed that the code could be run in a matter of microseconds and that the algorithm could be used without the robot having to stop. However, tests on the working robot were not so successful as the processing speed was hampered by external and timer interrupts necessary to keep the robot functioning.

The integrated test also revealed several unforeseen technical obstacles that are expanded upon in section 6.

3.3 System Testing Procedure

With all the subsystems deemed to meet the requirements, they were all brought together to form the whole robot. Testing for the robot commenced with first ensuring that the electrical and mechanical subsystems merged well together with everything in place. This was done by recreating the circuit with the various voltage regulators and resistors on a breadboard. The motors were then connected to the motor driver to ensure the integration was working. Then, to ensure the µC was connected properly, a simple code was loaded onto it to turn the indicator LED on. In this fashion, the system was tested in stages in such a way that each stage incorporated more features of the system. The different features tested were the push of a button to turn on the LED, turning on one motor (both forward and reverse), turning on both motors (both forward and reverse), ensuring that the.motors moved the robot, and so on.

Finally, once all the features of the system were proved to be operational, the next steps were to incorporate them all and begin the operation of the robot. Testing the Robot on the maze had 4 main aspects. These were ensuring that the robot was capable of going in a straight line while following the track, ensuring an accurate and consistent turn, performing U-turns at the end of a track and finally stopping at the black circle. The robot also had to follow the algorithmic logic. The algorithm used data from the sensors as well as an estimate of the distance traveled generated from a timer. It would log each junction as a new point recording its attributes and make a turning decision based on previous turns made. The robot's speed was measured at 15cm every 900ms and a timer interrupt was used to measure the time that passed between turns on the maze. The times were scaled so as to create an accurate measure which could be rounded to the lowest unit of distance on the maze(15cm). Each of these requirements were tested separately and only once they were deemed to meet the requirements, were they incorporated together.

3.4 System Testing Results

The systems were integrated successfully allowing for the implementation of the base operations. Line tracking was fine-tuned so that when the close sensors (S2 & S3 in Fig. 3) went over the line the corresponding wheel was slowed to a duty cycle of 85%. Next, the turning function was fine-tuned. A delay was added so that the robot would turn as its center of rotation was aligned with the track, then one wheel was put into reverse until one of the close sensors was triggered. This event would slow the turn until the front sensor (S1) hit the line at which point both wheels would resume moving forward. This slowing down allowed turns to be taken at speed without the risk of overshooting the line. Once the turning and line tracking was fully implemented it was safe to assume that if all the sensors were off the line the robot had reached a dead end. Checks were added using a timer interrupt to execute a u-turn if no lines were detected for a given time period. This function was also fine-tuned to prevent false triggers. Checks were added throughout the code so that when all sensors outputs were low the robot would stop and flash its LEDs as it would be safe to assume it was on the black circle.

The tests to measure distance were inconclusive as there were many factors that were hard to model predictably such as the non-linear acceleration of the motors and the robot cutting corners after becoming misaligned with the track. These factors made it impractical to use an algorithm relying on measured distances. This method was abandoned due to time and complexity constraints.

A second algorithm was trialed which navigated the maze by a preference of turns, If given options it would choose them in the following order: Left, Forward, Right, Back. It would explore until finding the end circle at which point it would iterate through a record of its turns taken and remove redundant sequences of turns relating to dead ends and loops. This method showed promise to optimise the route although it would not have guaranteed finding the shortest path. Unfortunately, due to time limitations this algorithm failed to deliver when testing its ability to optimise the path effectively.

4 Commentary on the Mechanical Design

The mechanical design can be split up into two sections; the driving mechanism, and the driven section. The driving mechanism consists of the motors and how they operated, including the turning mechanism. The driven section involves the chassis design of the robot, the weight associated with the chassis and the integration of the chassis and circuitry such as the connections and placements.

All designs researched used a differential steering system similar to those used in cars. Differential steering involves a set of wheels being able to turn at different rates. In a car, this is obtained through a differential, but in the designed robot, it was achieved through the use of a motor driver. This had the largest effect on the turning of the robot. Some previous designs had a system of slowing down one of the motors [4],[5]. This method has the advantage of not having to worry about the momentum of each wheel as much. This method chosen was not this method however. The method of putting one wheel forward and the other in reverse was chosen as it allowed the center of rotation to be controlled and situated in the middle of the two wheels. This method proved most beneficial to the setup and design of the robot as it allowed sharp corners which prevented misalignment of the robot after the turn. The setup and design of the motors were satisfactory and completed the task as desired.

The altering of the gearbox in each motor proved beneficial. The robot was able to drive at a quicker speed than that of an unaltered motor set. On top of that, the torque was still enough to not give any

problems. If the gearbox was altered any further, the speed would increase. This could have been desired but it would have put a strain on the processing power of the microchip.

The chassis of the robot was designed to be simple, lightweight and easy to alter. It was a simple single-layered platform with the intention of holding all circuitry on it. This proved not the case as the circuitry was larger than initially imagined. This resulted in not having enough space to fit everything in a single layer. What was then added on post-design was a mount to sit on top of the motor mounts such that the main veroboard circuit with the microchip could sit upon it. This was achieved with the use of spare laser-cut hardboard, 4mm thin dowels to ensure a tight fit and some bolts and nuts. This modification worked well and was able to free up some space for the remainder of the parts. The sensor circuit boards were then stacked on top of each other with the use of bolts and nuts such that two boards took the space of one board. This can be seen in the detailed drawing found in the appendix. An original designed image of the robot and a post-design modified image can be found in the appendix as well.

On the chassis, there were no designed slots for the sensors to be installed and so they were installed with Prestik. This was both beneficial and detrimental. The beneficial part of this design is that the sensors count be mounted quickly and could be positioned in any which way. The detrimental part of this design was that the sensors were often not straight and were sensing the ground at an angle. This could have been fixed if a predefined slot had been cut into the chassis for the sensors to fit. However, it would have compromised the maneuverability of the sensors. The same Prestik fit was also used for the sensor boards which posed the same issues. In the end, a predefined slot would have been more beneficial but did not subtract from the functionality of the robot at all. Therefore the design was satisfactory.

5 Commentary on the Electrical Design

The initial electrical design of the µC circuitry was to have the bare basics of the system user requirements (That is, one LED and one Button). It was soon realised that this made it very difficult to test and debug whether the system was performing as expected at different nodes in the maze. This was especially difficult when the robot was in operation on the maze and didn't perform as expected. This then made debugging the code much more difficult as we found ourselves not entirely knowing what was happening.

The solution to this problem was to add more LEDs that would serve as indicators as to what the robot was doing at a specific time. There were in total 4 additional LEDs added to the design, and this made the total number of LEDs on the central board 5. This made it much easier to test various functions of the system. An example of this would be in testing the line following and turning functions of the robot. Two of the LEDs would be set to turn on to indicate the robot is taking a turn as well as which direction it is turning, and the other 3 LEDs would be set to turn on to indicate a track being sensed.

Power consumption and total current drawn with all the incorporated circuitry was a major concern. The two regulators used in the circuit powered every component in the robot. The regulators, therefore, were at risk of being overloaded or consuming too much power based on the input voltage and the current drawn from the components. The maximum current into the µC power pins is 120mA as per the datasheet. This current is coming from the 3V3 regulator which also powers the TTL logic in the sensor circuitry. Using two different regulators, one for the µC circuitry and the other for the sensor circuitry would have reduced the risk of overloading the regulator.

Further improvements to the design would be to include more than just the μC chip on the main board. In the early design, it was mutually agreed not to include the ST-Link debugger on the main board but the act of transferring the chip from the robot to the development board and back each time new code needed to be flashed was very tedious and the time spent doing so would have been better spent in other developments. As such, including the debugger on the main board would have been very beneficial. Additionally, the initial design lacked a reset button. This meant having to disconnect the power source to the robot every time it was desired to restart the robot. This constant on and off could be detrimental to the components and is generally frowned upon in practice. Therefore, it was learnt to always include the reset button in future projects running the STM32F051C6 μC.

Our initial sensor placement was difficult to manage as it allowed the robot to go too far off track without triggering the Left or Right Correcting sensors resulting in the robot veering off the maze. It was decided to move these two sensors as close to the Forward sensor as possible and ensure they were placed at positions just wider than the width of the black tape. This ensured that when the robot was going slightly off track the sensors would trigger.

6 Commentary on the Systemic (Algorithmic) Design

This algorithm was created and tested using keyboard inputs. It proved highly effective navigating and solving the practice maze provided without any missteps.

However, when testing this code on the functioning robot it became clear that this solution was far less effective in practice. The main problem was that the algorithm required accurate values of distance. Two methods were considered to track distance. The first was to track wheel turns using a line sensor and electrical tape on the wheels. By using tape at a fixed interval it would be possible to measure distances as a fraction of the wheel's circumference. The second method was to use the STM’s internal clock to estimate the distance traveled by looking at the time. The external crystal oscillator on the robot would allow us to get accurate times however this strategy assumed that there would be abrupt acceleration and that the velocity of the robot would be somewhat constant when traveling on a straight line. It was decided that the second option would be more accurate and easier to implement.

When testing this algorithm on the completed robot it became clear that it would be problematic. Tracking accurate distances proved difficult even when allowing for a large margin of error when comparing two coordinate points. On top of this, the algorithm assumed that the correct path was always taken and did not account for mistakes. If one such mistake occurred due to ambient light or misalignment with the track due to a sequence of tight turns the algorithm would fail to complete the maze. Given time limitations it was decided to scrap this algorithm.

The final algorithm demoed made turns based on a preference of left, forward, right, back allowing it to go in circles and find the end of the maze. The sequence of turns made was recorded and then optimised using an algorithm that spotted redundant sets of turns such as exploring a dead-end or going in a circle. For example, if a Left -> Turn Around -> Left was spotted it would be removed and replaced with a Straight.

This Algorithm successfully found the end of the maze and stopped however due to time limitations the optimisation was poor.

Ways to improve these results could be using a wheel encoder to accurately measure distances traveled and using trigonometry to account for the robot's nonlinear path as it swerved to track a line. These steps could have allowed for a proper implementation of the first algorithm however the lesson learnt is that a less efficient but simpler algorithm would have been more practical to implement given the restraints on time and finances.

7 Conclusion

This project set out to design, build and implement a line following robot that entailed a mechanical design, an electronic design, and an algorithmic design. Each was designed separately and subjected to individual tests to ensure they worked accordingly. They were then consolidated to produce the final robot.

During testing of the robot, various issues arose and were dealt with accordingly. These issues varied from miscalibrated sensors to unstable movement of the robot. Progress in implementing and improving its operation was slow due to this but through consistent efforts, results were seen. The robot came from barely following a line to accurately traversing the maze from starting at any point and finding the end.

Unfortunately, however, we were unable to optimise the algorithm to find the shortest path in time for the demonstration but we believe we were on the verge of accomplishing that. In retrospect, this could have been achieved by having multiple ideas for finding the shortest path by conducting more research and being more prepared to deal with the problems we encountered.

At the beginning of the project, the task assigned seemed daunting and unattainable due to a lack of knowledge, experience and self-assurity. Over the course of the project, these concepts that seemed too difficult at first became just another obstacle to climb. As the project continued, knowledge, experience and self-assurity was acquired. At the end of the project, it was learnt that a task that may seem too big was just another task not yet tried and that most projects can be done, if given the correct motivation and duration.

8 References

[1]Elonics.in. *IR (InfraRed) proximity sensor / Obstacle detector circuit using LM358 opamp - Elonics*. [online] Available at: http://elonics.in/breadboard-projects/infrared-ir-proximity-obstacle-sensor-using-lm-358 [Accessed 12 Aug. 2019].

[2]S. Sakib, A. Chowdhury, S. Ahamed and S. Hasan, "Maze solving algorithm for line following robot and derivation of linear path distance from nonlinear path", 16th Int'l Conf. Computer and Information Technology, 2014. Available: 10.1109/iccitechn.2014.6997314 [Accessed 2 October 2019].

[3]"Dijkstra's Algorithm in C - The Crazy Programmer", *The Crazy Programmer*, 2019. [Online]. Available: https://www.thecrazyprogrammer.com/2014/03/dijkstra-algorithm-for-finding-shortest-path-of-a-graph.html. [Accessed: 03- Oct- 2019]

[4]V. HYMAVATHI and G. VIJAY KUMAR, "DESIGN AND IMPLEMENTATION OF DOUBLE LINE FOLLOWER ROBOT", *International Journal of Engineering Science and Technology*, vol. 36, no. 0975-5462, pp. 4946-4952, 2011.

[5]KATHMANDU UNIVERSITY, "LINE FOLLOWING ROBOT", KATHMANDU UNIVERSITY, KATHMANDU, 2016.

9 Appendix

9.1 Sensor diagram and Veroboard Design

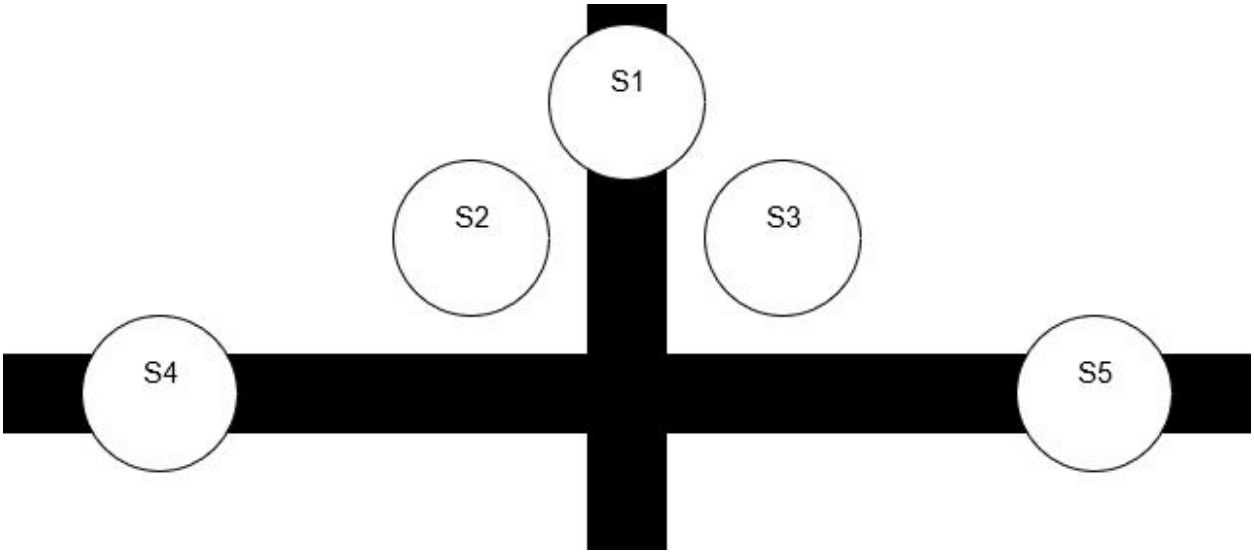


Figure 3: Sensor diagram

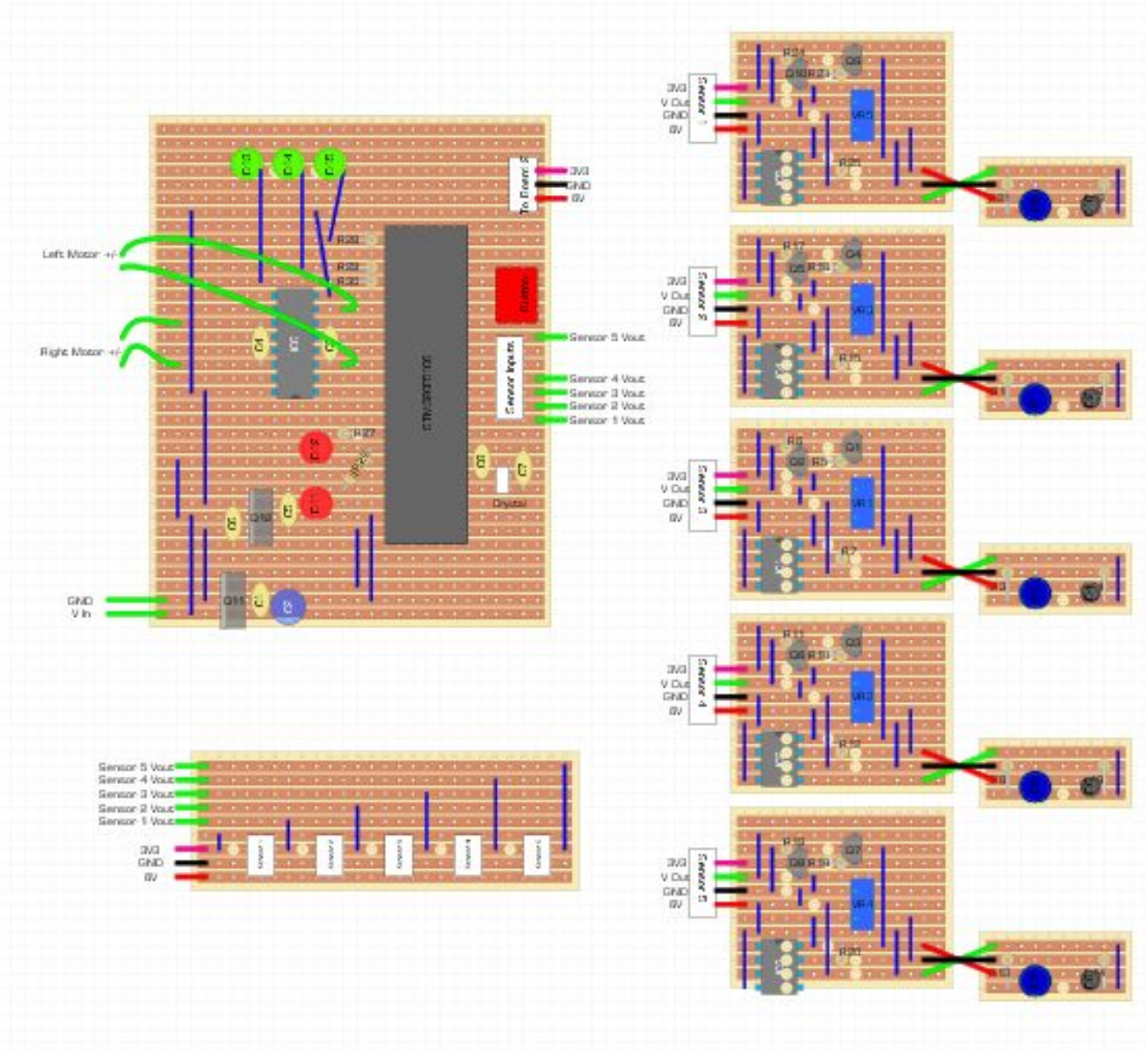


Figure 4: Circuit design of the sensors and main board

9.2 Schematic of the Final Design

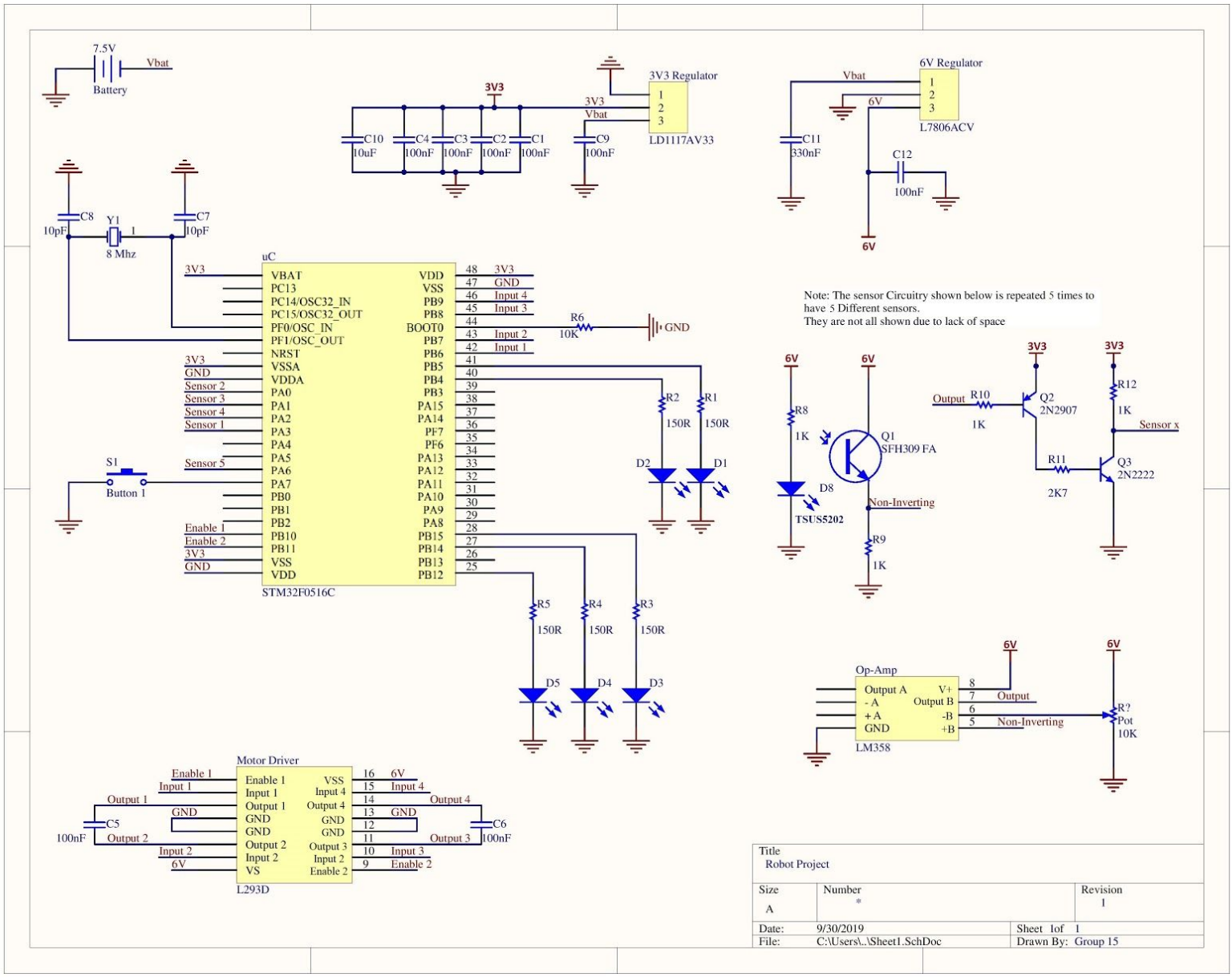


Figure 5: Design Schematic

9.3 Rendered images of the designs

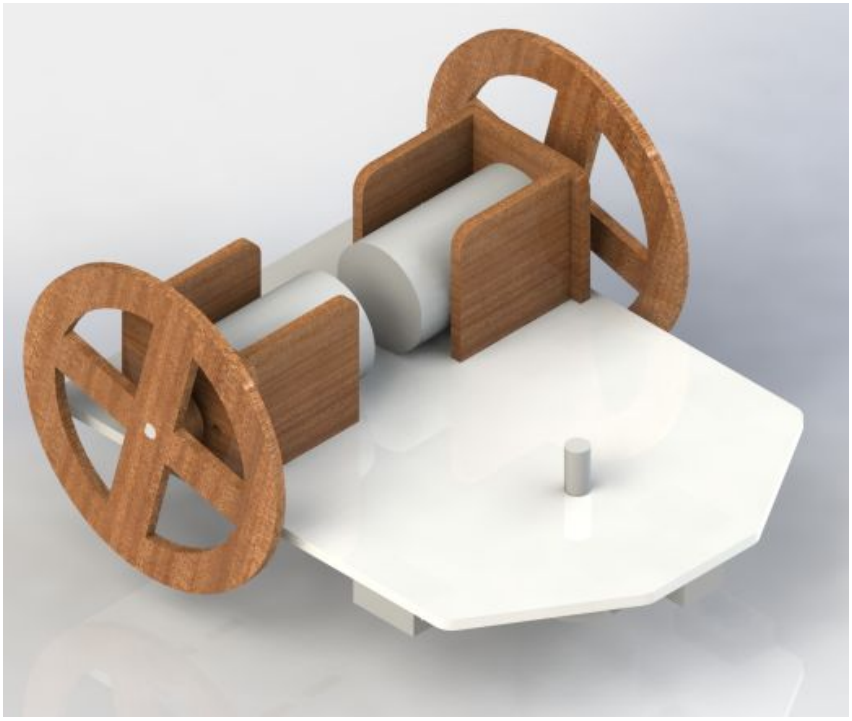


Figure 6 The robot with the original design implemented

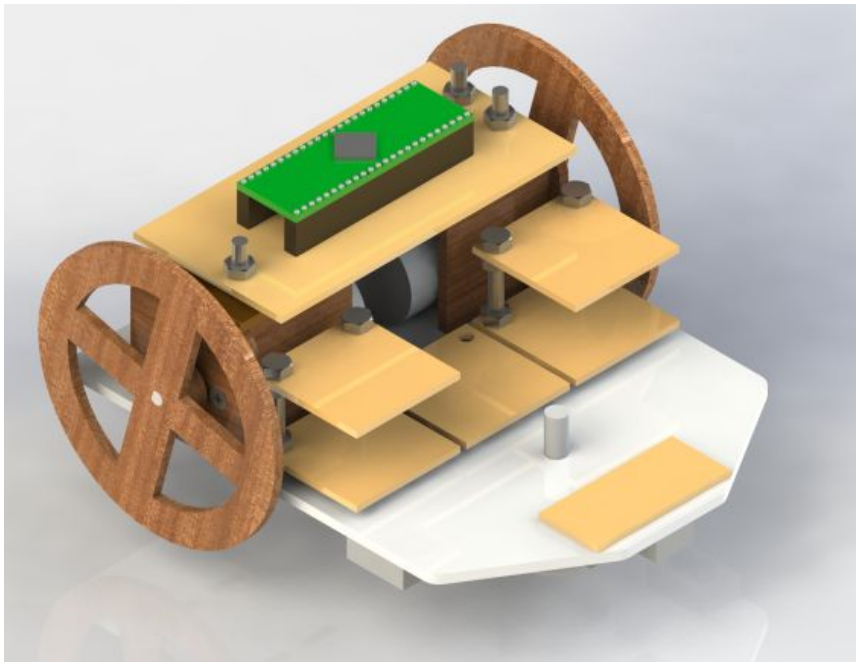


Figure 7 The robot with the post-design modifications implemented

9.4 Final Mechanical Drawing