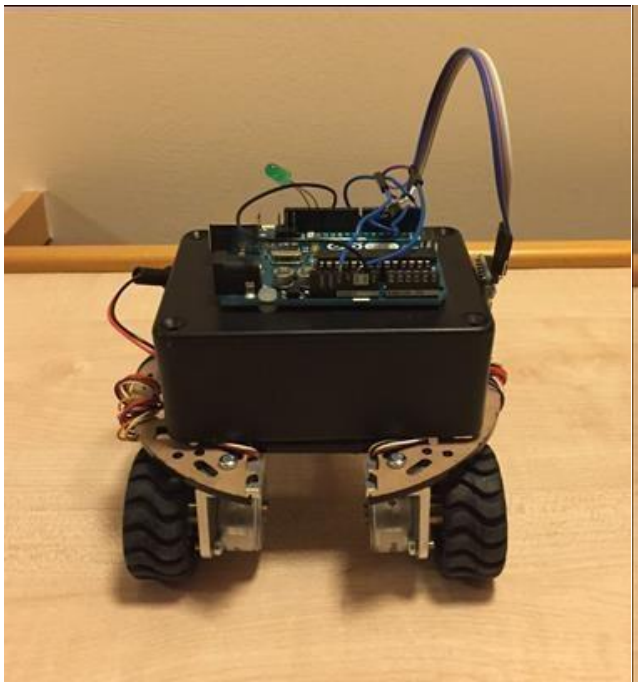
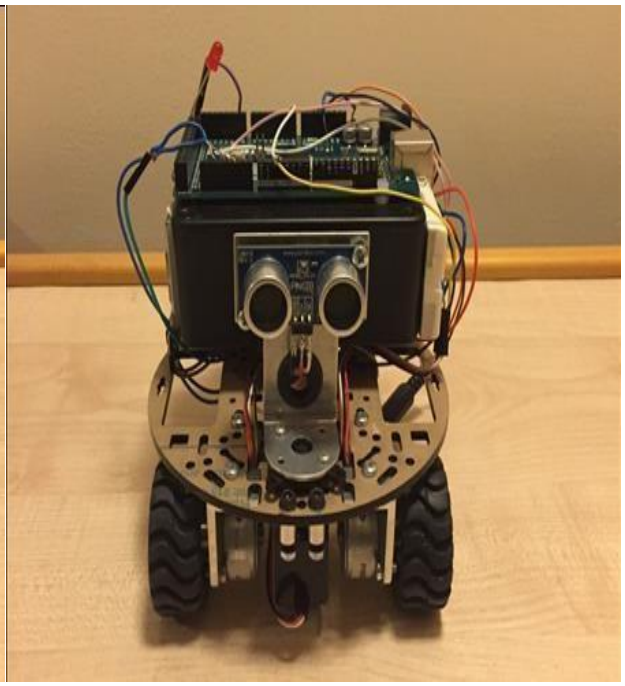
Maze Solving Robot

Design Project

May 2016

Abstract:

This thesis presents the development of two autonomous robots that are able to navigate through a maze. The aim of the project was achieved by programming the microcontrollers combined with a sensor, wireless communication, data logger and motor drive circuit. A Literature review was done to investigate the existing hardware and software devices to determine the optimum components to be implemented in the design. The project consisted of programming the main robot to solve a maze, map it, store the map and transmit it to a follower robot. This allowed the follower robot to use the instructions received to navigate through the maze. Both robots were designed and tested and were able to navigate through the maze. However, some challenges were encountered regarding the displayed map and the motors used; improvements are suggested.

Table of Contents

Table of Figures

1. Introduction:

1.1.Background:

Autonomous robots are robots that are able to perform tasks without the help of an operator. One popular autonomous robot is the maze solving robot, which is a small robot that has the ability to solve a maze, for example from the given starting point to the center in a short period of time. It became very popular to an extent that an internationally renowned competition called 'Micro-Mouse' was made for it. When the maze solving robot is placed in an unknown environment, it needs to investigate the surrounding area. For instance, if there is an obstacle in its way, it should execute appropriate actions such as avoiding it. Therefore, many paths finding algorithms were developed and implemented to help the robot solve the maze.

Around the 1950s, the robotic industries were looking into difficulties that arise from solving a maze and till today it is one of the main themes in robotics. One of the topics in the robotics is the decision-making algorithm where the robot requires a good algorithm to be used when the robot is placed in an unfamiliar environment. One of the areas of decision-making algorithm is a pathfinding algorithm. Throughout the history, different types of algorithms were developed such as wall follower, Flood Fill, A* algorithm and much more. In the 1970s, the old generation of maze solving robots was physically huge due to numerous microcontroller chips used in design to help in implementing the algorithms to the robot. As the technology is evolving, the shape of the design became smaller. Figure 1 and Figure 2 shows the evolvement of the robot.

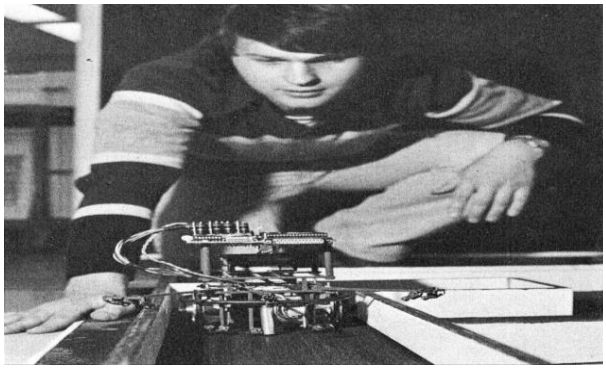


Figure 1: Old generation maze solving robot

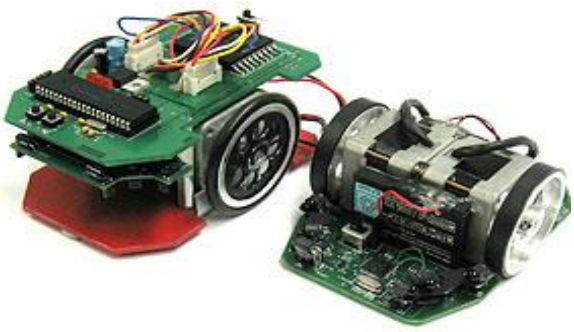


Figure 2: Modern Maze Solving robot

Also in the 1970s, a countless number of engineers was inspired by the area of autonomous robots that a competition called “Micro Mouse” was organized, it then became known worldwide. The main objective of this competition is that robots should solve a 16x16 grid wooden maze from a given starting point to the centre with a time limit of 10 minutes. Furthermore, the design of the robot should follow the required rules of the competition where length and width limit of the robot is 25cm, as there is no restriction on the height of the robot (Micromouse USA, 2016). Figure 3 displays a micromouse robot in a maze.

Figure 3: Micro Mouse competition in San Diego CA (IEEE, 2013)

1.2.Introduction to the project:

This project was chosen to develop two autonomous maze solving robots where the main robot has the capability to solve a given maze with the help of sensors, stores the map in an external memory and transmits it wirelessly. The follower robot has the ability to receive the map which it uses to solve the maze without the need to utilise its on-board sensors.

1.3.Motivation:

The Maze Solving robot is a well-known project chosen for its potential in improving robot design. First improvement task is enabling the first robot to transmit the map of the maze to another robot, the second robot receiving the map can then follow the directions that first

the robot made. This improvement in design could be used in several scenarios: in space explorations, where for example, the main robot is scanning an area and if an anomalous situation occurs causing a shut down or if the robot crashed, the second robot can follow the movements of the main robot and continue scanning the area; in rescue operations, where the main robot is scanning the area and the robot malfunctioned in the middle of an operation, the second robot can then take over follow the first's movements and can provide assistance to exit an obstacle

2. Project Specification:

The specifications of the project are divided into two parts:

2.1.Performance Specification:

The main robot must be able to:

1. Solve the maze and map it
2. Save the map in external memory
3. Transmit the map to the follower robot

The follower robot must be able to:

1. Receive the map of the maze
2. Follow the movements of the main robot

2.2.Design Specification:

The maze and robot designs must be constructed in a particular way:

1. Maze:

The maze must be designed to a certain grid cell consisting of entrance, exit, and dead end points.

2. Robot Layout:

The robots must be of a size allowing it to move and turn inside the maze.

3. Literature Review:

For this section, it outlines various existing technology to be implemented for this project. Throughout the review, it will answer the following questions:

1. How will the robot be able to move?
2. How is the data transmitted?
3. How will the robot detect an obstacle?
4. How will the robot be able to solve a maze?

3.1.Hardware components:

3.1.1. Motors:-

In this section, it will answer the question about the mobility of the robot. In the normal operating mode of the electric motor, the magnetic field of the motor and the current coming from the winding of the coils work together to generate a linear or rotational force (which named torque) inside the motor.

There are several different types of motors, which we will discuss in the following sections:

1. DC motor:

DC motors consist of two types: brushed and brushless DC motor. The brushed DC motor contains two magnets facing each other surrounding an armature. The rotation of the armature is caused by the magnetic field generated by the flow of the current that opposes the field of permanent magnets (Sabin Mathew, 2014). Figure 4 shows the construction of the motor and how it operates.

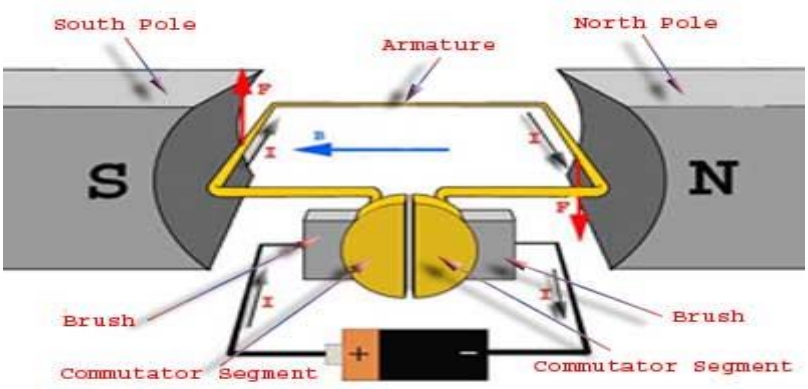


Figure 4: Construction and operation of Brushed DC motor

The brushless DC motor was developed in the near 1960s, as an improvement to the design of the brushed DC motor by replacing the brush with the winding of coils (Schweber, 2002). The motor consists of a rotor with permanent magnets and a stator with windings. It works in a similar manner to the brushed DC motor but the magnetisation of the windings could be controlled electronically. As the winding is magnetized, it causes the rotor to rotate due to opposing magnetic fields and as the winding are controlled electronically, the rotor can rotate in assigned pattern (Sabin Mathew, 2013). Figure 5 shows the construction and operation of the motor.

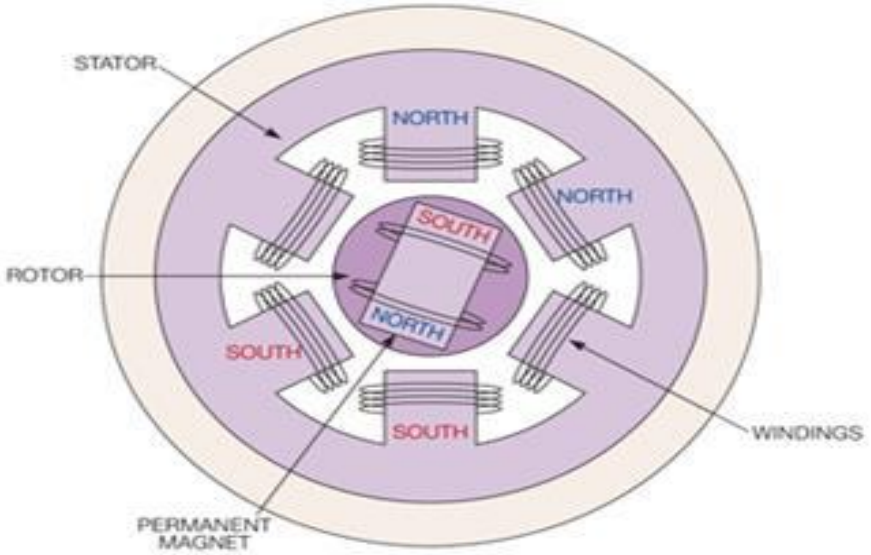


Figure 5: Construction and operation of Brushless DC motor

One of the types of brushless DC motor is a stepper motor.

2. Stepper motor:

A Stepper motor is an electromechanical actuator that converts a digital input signal into analogue motion by magnetizing a magnet causing the rotor to rotate to a fixed angle. Figure 6 shows the operation of the stepper motor. Its operation is similar to the operation of brushless DC motor.

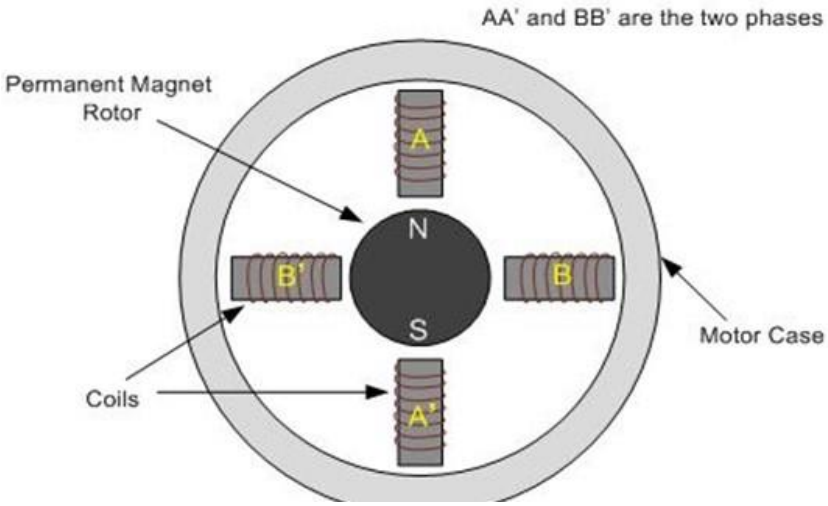


Figure 6: Stepper motor Diagram

Two-phase stepper motor has two winding arrangements:

I. Unipolar Stepper motor
II. Bipolar Stepper motor

The unipolar stepper motor has an extra wire placed at the centre of each coil; this position causes a partial rotation in its operational mode. Therefore, there is no need to reverse the polarity of the current to reverse the direction of magnetic field, for instance, to cause anti-clockwise rotation of the rotor let current can be allowed to pass from A-COM to A (2). Figure 7 shows construction and operation of the unipolar stepper motor.

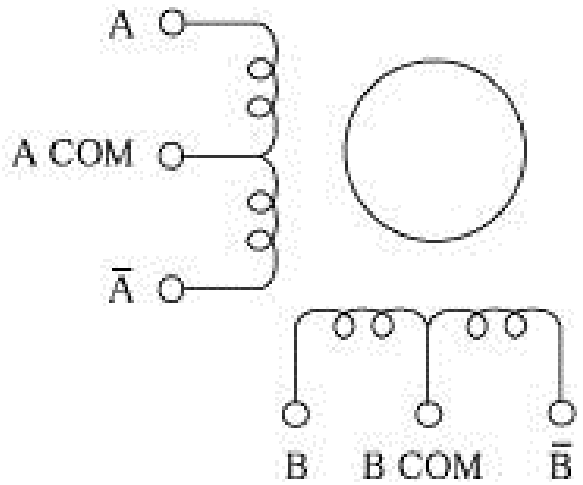


Figure 7: Unipolar Stepper motor

Bipolar stepper motor has one winding in each phase. In order to reverse the magnetic field, the polarity of the current should change. Figure 8 shows construction and operation of the bipolar stepper motor.

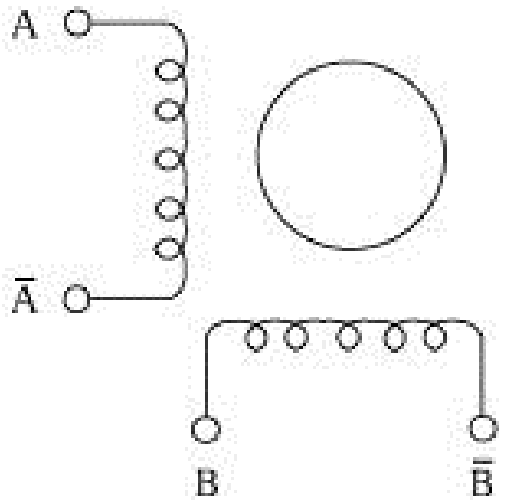


Figure 8: Bipolar Stepper motor

3. Servo Motor:

The Servo motor is a linear or rotatory actuator that can give precision control over the linear or angular movement of an object. The servo motor is made up of a motor (DC or AC), a position sensor, gearbox and controlling circuit. Figure 9 shows the closed-loop servomechanism. In this figure, the position sensor is connected to the motor shaft and the gearbox is used to reduce the speed of rotation of the motor to improve the torque. When there is no electrical signal; the sensor knob position stays constant. When there is an electrical signal, the input signal is then compared with error signal in an error detection amplifier and the difference between the signals is then fed to the motor causing it to rotate. When the motor rotates, it causes the position cog to rotate which generates an output feedback signal. When the angular position of the sensor changes it affects the output signal. The position cog continues to rotate until the output signal and input signal are equal then there will be no difference in the error detection amplifier ceasing the rotational movement of the motor (Electrical4U, 2020).

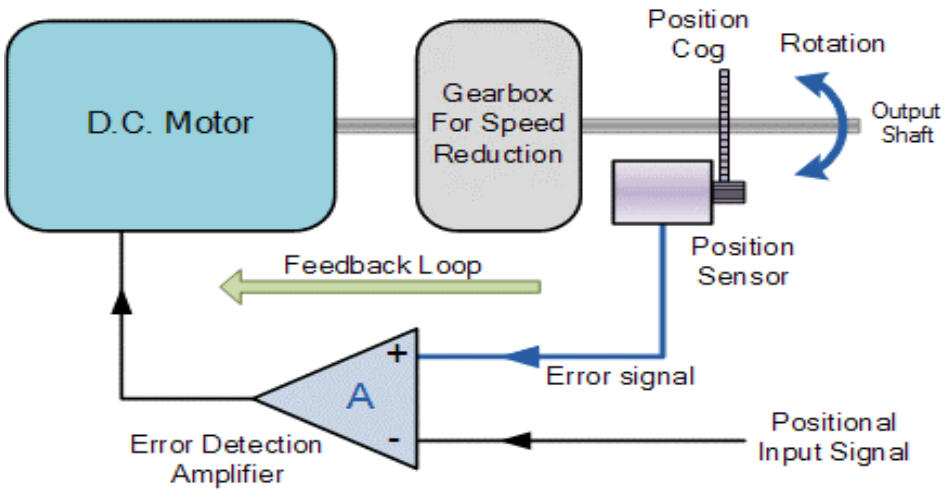


Figure 9: Closed-loop servo mechanism

3.1.2. Communications:-

This section will address the question of about data transmission. Communication on the Internet of Things (IoT) is a method of transmitting information between devices. The wireless communication in the IoT consists of several components:

1. Bluetooth:

Bluetooth is a wireless telecommunication technology that transmits data between fixed or mobile devices. A Bluetooth device needs to a have transceiver chip to be able to transmit and receive data at a frequency of 2.45GHz. The Bluetooth module consists of 40 channels 37 of which are data channels that transmit data once the devices are connected. The remaining 3 are advertising channels that discover and connect devices together (Joh, Yang and Ryoo, 2016). Figure 10 shows the frequency and the channels in Bluetooth module.

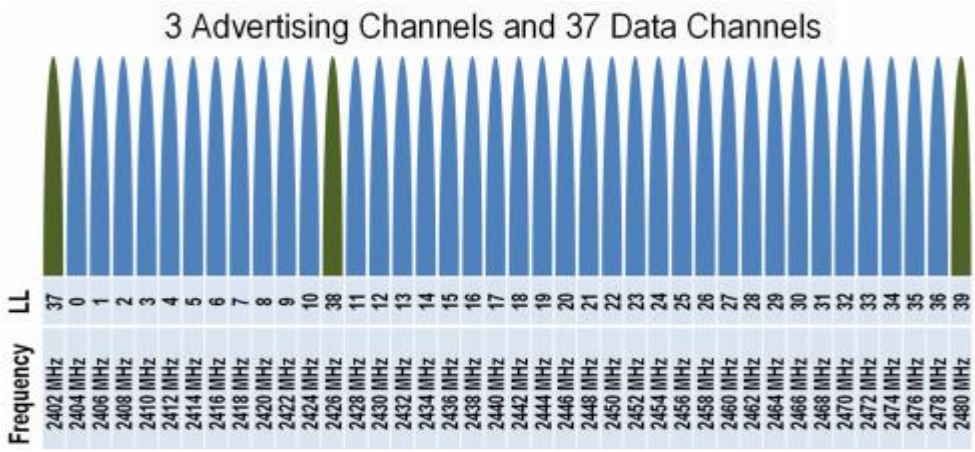


Figure 10: Frequency and the channels in Bluetooth module

2. Wi-Fi:

Wi-Fi is another wireless telecommunication technology that is capable of transmitting data between fixed or mobile devices by using radio waves. The electromagnetic field is created by supplying a radio frequency (RF) current to an antenna; this field is then dispersed throughout space connecting devices through an access point (AP). Wireless network adaptor assists the devices connection to the access point.

3.1.3. Sensors:-

To recap, the sensors answer the question about obstacle detection. Sensors are devices where it is able to convert specific input for example light, ultrasonic and much more; to an electrical signal for a system to record, measure and execute a required action if there were any abnormalities in the signal.

There are multiple types of sensors that we can use for our main robot to sense its environment:

1. Infra-Red (IR) Sensor:

The IR sensor detects a difference in the infra-red wavelength between ranges of 700nm to 1um. The sensor is composed of an IR LED and an IR LED receiver. It operates by using the IR LED to send an infra-red wave to an object where the colour of the object affects the absorption of infra-red wave and the receiver is used to receive the incoming reflection of the wave. The infra-red wave is converted to an electrical signal which is then fed to the inverting input port of the comparator which then compares this signal with signal entering at the non-inverting port of the comparator. The output difference at the comparator would cause the LED to turn on or off. Figure 11 show the principal of the IR sensor and Figure 12 shows the circuit module of the LED comparator (Mohammad, 2009).

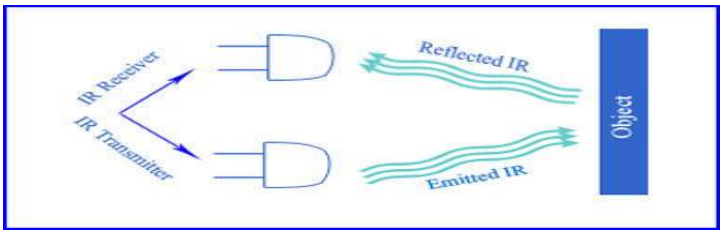


Figure 11: Principal of the IR sensor

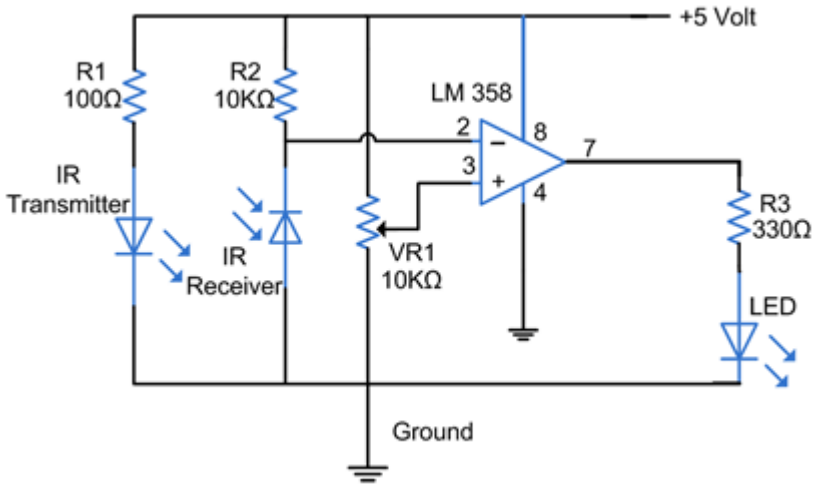


Figure 12: Circuit module of the IR LED comparator

2. Ultrasonic Sensor:

Ultrasonic sensor is a sensor that has a capability to detect the reflection of an ultrasonic burst. The sensor is made up of an ultrasonic emitter and an ultrasonic receiver. The principal of the sensor is that the emitter releases an ultrasonic burst towards an object and the receiver receives the reflection of the ultrasonic burst. Then the time for the receiver to receive that reflection is then calculated to get the distance from the sensor to the object (Mohammad, 2009). Figure 13 shows the principal of the ultrasonic sensor.

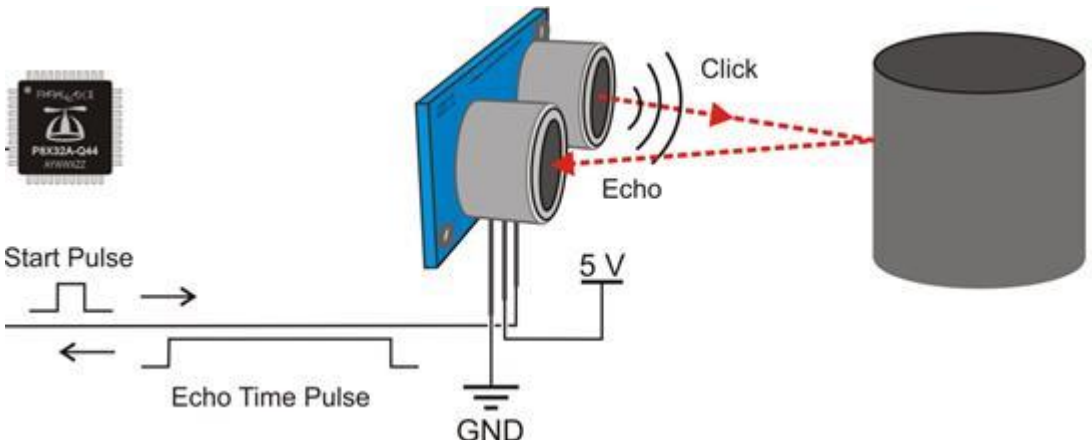


Figure 13: Principal of the ultrasonic sensor

3.2.Pulse Width Modulation:

For an analog device to turn on for a specified period of time a high digital signal is needed. Therefore, Pulse Width Modulation (PWM) is used to modulate the pulse width of the digital signal where the period of keeping signal high is called duty cycle.

For an analog device to turn on at 5V for a specified period of time a high digital signal is needed. By controlling the time period, it varies the amount of voltage flowing to the device. This period is called percentage duty cycle. Therefore, with a duty cycle of 50% the device receives 2.5V (SparkFun, 2013). The PWM is used in driving motors due to its high efficiency in power reduction. Figure 14 shows PWM with various duty cycles.

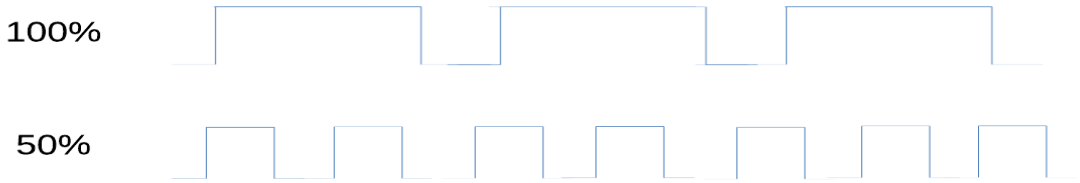


Figure 14: PWM with various duty cycles

3.3.Algorithms:

3.3.1. Wall follower Algorithm:-

The Wall Follower Algorithm is a code used for a robot to solve a maze by using a wall as a reference. The robot follows either the left or right wall to solve a maze. The algorithm collects the input from the sensor and compares it with set priority conditions and then executes a movement (Gupta and Sehgal, 2014). Table 1 shows the priority conditions for both left and right wall follower algorithms. Figure 15 shows an example of left-hand wall follower algorithm (BakarSayutiSaman and Abdramane, 2013).

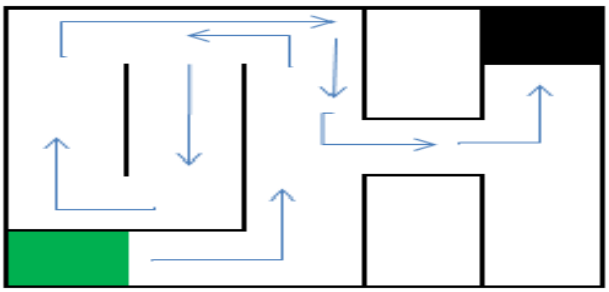


Figure 15: Left-hand wall follower

Table 1: Conditions of Wall follower algorithm

Left wall follower algorithm:	***Right wall follower algorithm:***
If you can turn left, then turn left	If you can turn right, then turn right
If you can't turn left , then move forward	If you can't turn right, then move forward
If you can't move forward, then turn right	If you can't move forward, then turn left
Else turn around	Else turn around

3.3.2. Flood fill:-

Flood fill algorithm is also another technique used to solve a maze by comparing the cell values. The algorithm is based on "Bellman-Ford Algorithm" where this algorithm is more complex and able to solve any maze. The maze is sorted into a multidimensional array where the code fills the cells with values, and then the code compares the values assigned to each cell in the maze. These values indicate the number of steps needed for the robot to reach the end or centre of the maze (Gupta and Sehgal, 2014). Figure 16 shows an example of flood fill algorithm (Gupta and Sehgal, 2014).

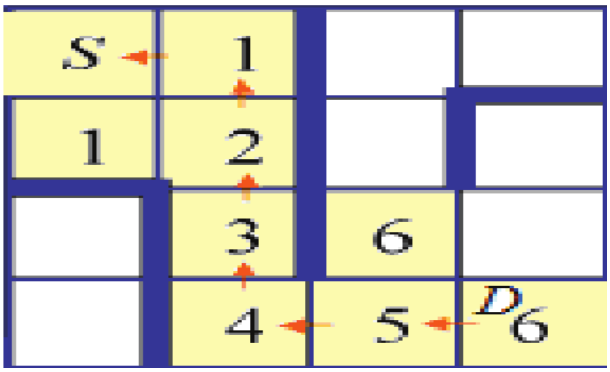


Figure 16: Bellman-Ford Algorithm

Table 2: How flood fill algorithm performs

Flood fill algorithm:
Update the wall map
Flood each cell with new distance values
Compare each neighbouring cell around the robot and determine which cell with lower distance value
Move the robot towards the cell with lower distance value

3.4.Assessment of current methodologies and techniques:

From the articles that are mentioned in both hardware and software component sections, there are advantages of using ultrasonic sensors than IR sensors in obstacle detection (Mohammad, 2009; Kassim *et al.*, 2013):

1. Ultrasonic sensors have better obstacle detection resolution in shorter distances than IR sensors.
2. Ultrasonic sensors are capable to detect obstacle and is not affect by any change on surface reflectivity.

Whereas, when using motor to mobilise the robots, the advantages of using stepper motor over servo and DC motors (Helen, 2019):

1. Stepper motors have greater precision control of the movement of the robot than servo and DC motors.
2. Stepper motors have higher torque for moving the robot at lower speed than servo and DC motors.
3. Stepper motors are easily controlled using micro-controllers than servo and DC motors.

Furthermore, in terms of connectivity of the two robots the Bluetooth was selected (Bhatt, 2011):

1. Bluetooth module is less complex for coding than Wi-Fi.
2. Bluetooth module is capable to perform direct connectivity between modules than Wi-Fi.

When it comes to the software algorithms, it involves using obstacle detection algorithms which can be done using sensors. Algorithms mentioned in section 3.3 have their advantages (Gupta and Sehgal, 2014):

1. Wall follower algorithm is not complex like Flood-Fill algorithm.
2. Wall follower algorithm is capable of delivering assured solution of solving maze than Flood-Fill algorithm.
3. Wall follower algorithm implementation on robot is easy than Flood-Fill algorithm.

4. Hardware and Software Description:

This section consists of hardware and software components that are used to achieve the specifications of the project.

4.1. Hardware Description:

4.1.1. Arduino UNO:-

Arduino UNO is a microcontroller board that runs on an 8-bit microcontroller chip "ATmega328P" and it contains several features, for instance; a 5V voltage regulator that regulates voltage, a USB port, an ATmega16U2 chip that is used as a USB to serial converter, a power jacket, a reset button, a crystal oscillator that resonates at a frequency of 16MHz allowing the Arduino to control the duration of the signal pulse that is going to either the analog or digital port. These features maintain the functionality of the microcontroller (Arduino, 2022b). Figure 17 shows a picture of an Arduino UNO board. Table 3 shows important characteristics of the board.

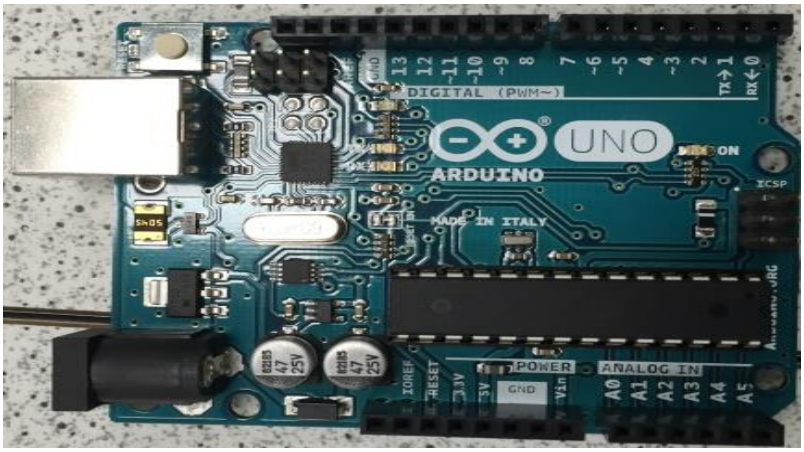


Figure 17: Arduino Uno microcontroller

Table 3: Characteristics of the Arduino UNO board

Microcontroller chip	ATmega328P
Operating Voltage	5V
Recommended Input Voltage	7-12V
Limited Input Voltage	6-20V
Digital I/O Pins	14(6 of them are PWM Pins)
PWM I/O Pins	6

Analog Input Pins	6
DC Current per I/O Pin	20mA
DC Current for 3.3V Pin	50mA
Flash Memory	32KB
SRAM	2KB
EEPROM	1KB
Clock Speed	16 MHz
Weight	25g

4.1.2. Arduino MEGA:-

The Arduino MEGA is another microcontroller board that runs on an 8-bit microcontroller chip "ATmega2560"; it contains similar features to Arduino UNO and the only difference being the increase in the amount of digital and analog ports (Arduino, 2022a). Figure 18 shows an image of an Arduino MEGA. Table 4 shows important characteristics of the board.

Figure 18: Arduino MEGA board

Table 4: Characteristics of the Arduino MEGA board

Microcontroller chip	ATmega2560
Operating Voltage	5V
Recommended Input Voltage	7-12V
Limited Input Voltage	6-20V
Digital I/O Pins	54(15 of them are PWM Pins)
PWM I/O Pins	15
Analog Input Pins	16

DC Current per I/O Pin	20mA
DC Current for 3.3V Pin	50mA
Flash Memory	256KB
SRAM	8KB
EEPROM	4KB
Clock Speed	16 MHz
Weight	37g

4.1.3. PING))) Sensor:-

PING sensor is an ultrasonic sensor with the frequency of 40 KHz tone. It uses a microcontroller to send a high input pulse to the PING))) sensor so that it starts measuring the time it receives the echo, then the sensor waits for the microcontroller to start its PULSIN command. It then emits the ultrasonic burst as well as a high output pulse to the microcontroller. When the sensor receives the echo it converts the high output signal to low and stores the period the signal stayed high in PULSIN command. Lastly, it uses the speed equation to calculate the distance of an object by equating the speed to the speed of sound in air (Parallax, 2009, 2013; Kassim *et al.*, 2013). The speed equation is shown below.

$$V = \frac{d}{t} \text{ as } c = 343 \frac{m}{s}$$

$$d = c \text{ x } t = 343 \text{ x } t$$

Figure 19 and Figure 20 shows the PING))) Sensor and its principle. Also, Table 5 contains the characteristics of the PING))) Sensor.

Figure 19: PING))) Sensor

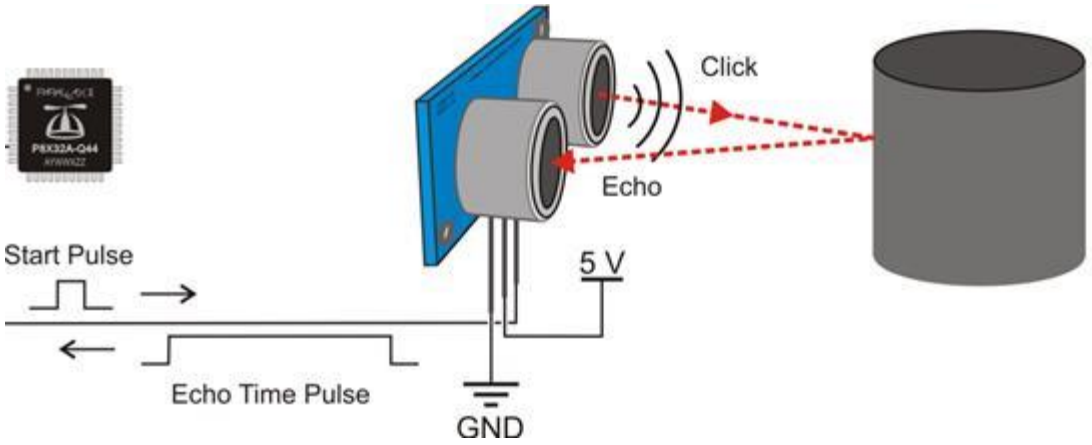


Figure 20: Principal of the ultrasonic sensor

Table 5: Characteristics of the PING))) Sensor

Range	3cm to 3.3m
Operating Voltage	5V

4.1.4. OpenLog:-

SparkFun OpenLog is a serial data logger that is able to store data in the form of a text file (.TXT) on a microSD card. The data logger supports a microSD card with a memory capacity of 16MB to 64GB and file type of FAT16 or FAT32 (Nathan Seidle, 2014). Figure 21 shows the OpenLog and Table 6 shows characteristics of the data logger.

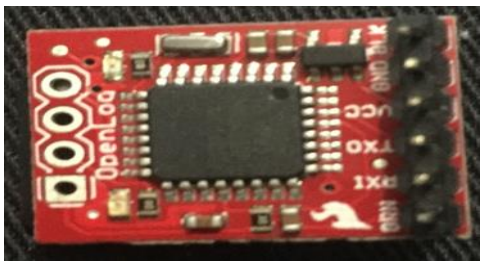


Figure 21: SparkFun OpenLog

Table 6: Characteristics of the data logger

Microcontroller	ATmega328
Operating Voltage	3.3V to 5V
Baud Rate	Up to 115200bps
Idle Current	2mA
Maximum Current	6mA

4.1.5. HC-05 Bluetooth module:-

HC-05 Bluetooth module is a wireless transceiver module that uses radio frequency (RF) to transmit data between devices at a frequency of 2.4GHz. This module is able to act as either master or slave. The term master means that the module can transmit data and slave mean that module can receive data (iteadstudio, 2010). Figure 22 shows HC-05 Bluetooth module and Table 7 shows characteristics of the module.

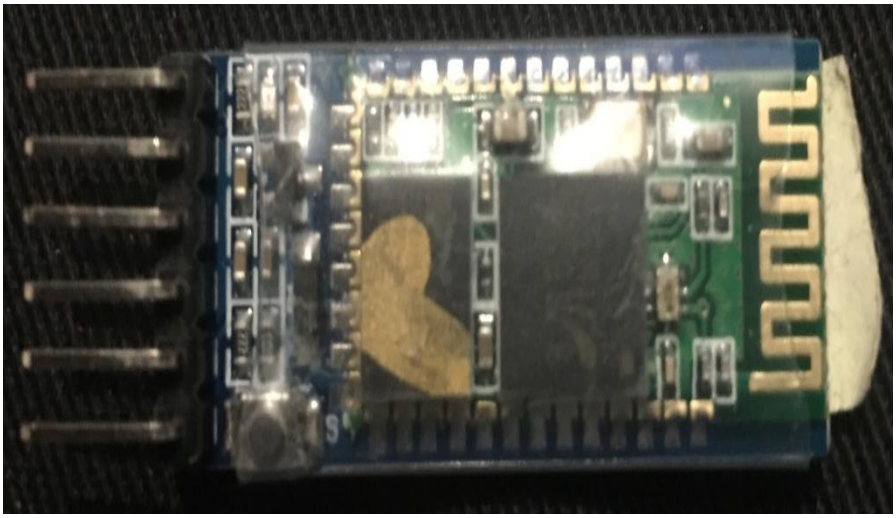


Figure 22: HC-05 Bluetooth module

Table 7: Characteristics of the Bluetooth module

Frequency	2.4 GHz
Operating Voltage	3.3V
Mode	Can be either Master or Slave
Baud Rate	Up to 1382400bps

4.1.6. Unipolar Stepper Motor:-

As explained in section 3.1.1, the unipolar stepper motor is an electromechanical motor that converts an input digital pulse into analogue motion. The motor magnetizes the coil by running current through it; the magnetic field of the coil interacts with the magnetic field of the magnet in the rotor causing the rotor to rotate. The motor uses the PWM to drive the motor to allow it to rotate in an assigned pattern. Figure 23 shows motor used and table 7 shows characteristics of the motor.

Figure 23: Unipolar stepper motor

Table 8: Characteristics of the selected stepper motor

Manufacturer Part Number	PM25L-048-15
Rated Voltage	12V DC
Rated Current	100mA
Rated Torque	50 g.cm
Step Angle	7.5^0
Weight	31 g

4.1.7. Servo Motor:-

As explained in section 3.1.1, the servo motor is a linear or rotational actuator that gives the precision control the linear or angular motion of an object. The motion of the servo motor is also controlled by using PWM and this servo motor is a positional rotation motor that is combined with a sensor for the purpose of moving the sensor to scan the area from 0^0 to 180^0 (Helen, 2019). Figure 16 shows the servo motor and table 8 shows characteristics of the motor.

Figure 24: Servo motor

Table 9: Characteristics of the selected Servo motor

Operating Voltage	5V DC
Rated Torque	38 oz.in
Angular Motion	Can rotate from 0^0 to 180^0
Weight	44 g

4.2. Software Description:

4.2.1. Arduino IDE:-

Arduino Integrated Development Environment (IDE) is a program that uses the C programming language to write a code which can be uploaded to Arduino board. The C programming language is a series of step by step instructions that are converted into machine code which is then transferred to the Arduino board through a USB cable. Moreover, the program also contains a serial monitor which displays the result that the microcontroller has collected from its input and output ports (Arduino, 2018). Figure 25 shows the Arduino IDE and Arduino Serial monitor. Additionally, the Arduino IDE is an open source program where people can freely share their ideas or codes and can obtain example codes and libraries by accessing the Arduino website.

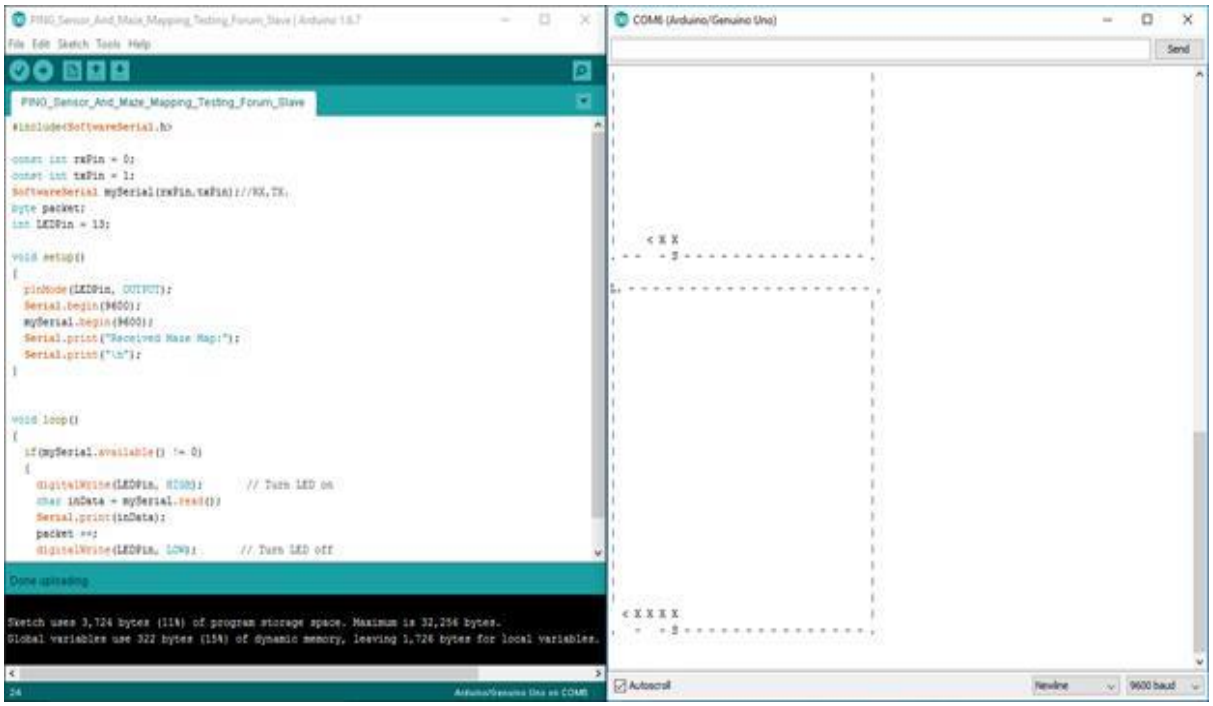


Figure 25: Arduino IDE and Arduino Serial monitor

4.2.2. Fritzing::-

Fritzing is a program that can display a graphical representation of the circuit design in the form of breadboard, schematic, and PCB. The libraries in the program consist of graphical design of the components such as microcontroller boards, motors etc. It is also an open source program where programmers can share their design of the circuit, develop new PCB component and code through the fritzing website (Nawale, 2020). Figure 26 shows the Fritzing software.

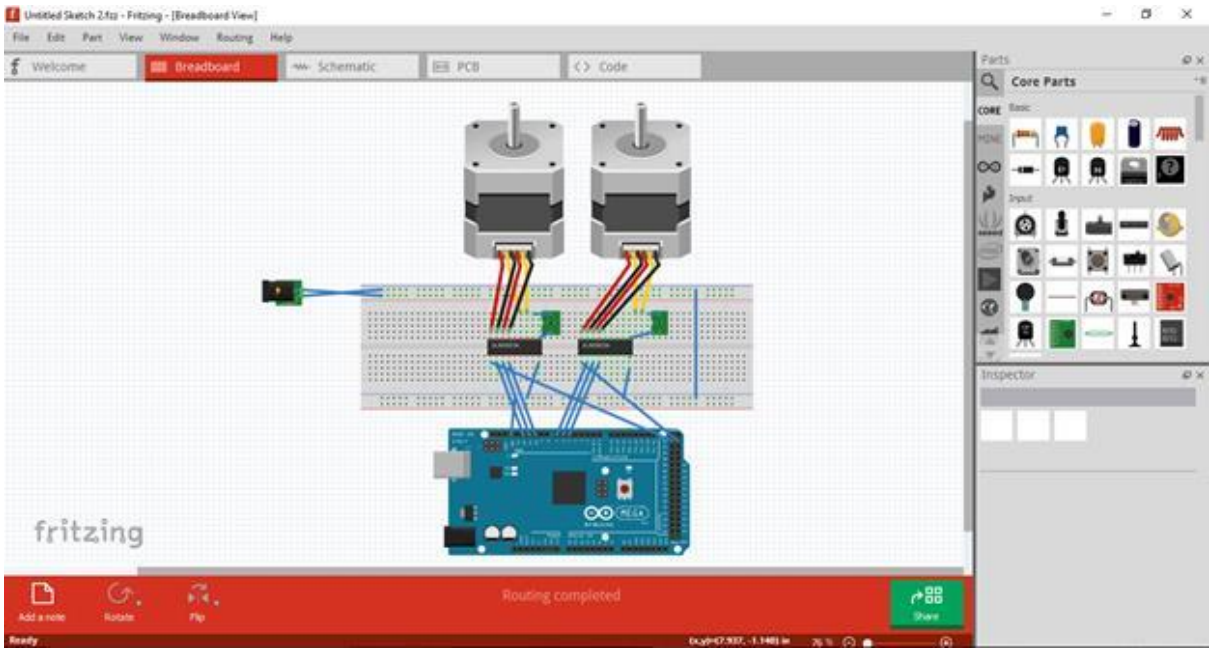


Figure 26: Fritzing software

5. Design Methodology:

The following sections consist of methodologies used in designing the robots.

5.1. Software Methodology:

5.1.1. Simulation Block Diagram:-

Figure 27 represents the block diagram of the simulation result of the main and follower robot program. The block diagram shows how the results are transmitted from the Arduino board to PC and shown in the Arduino serial monitor. By uploading the program to the Arduino board with the help of USB cable the Arduino board in the main robot, for instance, can produce maps of the maze, results of the scanning distances, movement and updated map of the maze and transmit it to the serial monitor for the operate to see the result.

Figure 27: Block diagram of simulation results

5.1.2. Scanning Method:-

When the main robot enters section of the scanning code, the robot uses the servo motor to rotate the PING))) sensor to record the distance between it and the wall then the motor returns back to its initial position. The code below displays how the robot scans the maze.

```
// Scanning the Perimeter //
// Scan Left //
  servo.write(180);
  delay(2000);
  leftscan = Sensor();
  leftscan = inches;
  Serial.print("LeftScan: ");
  Serial.println(leftscan);
// Scan Center //
  servo.write(90);
  delay(2000);
```

```
  centerscan = Sensor();
  centerscan = inches;
  Serial.print("CenterScan: ");
  Serial.println(centerscan);
// Scan Right //
  servo.write(0);
  delay(2000);
  rightscan = Sensor();
  rightscan = inches;
  Serial.print("RightScan: ");
  Serial.println(rightscan);
// Return To Center //
  servo.write(90);
  delay(5000);
```

5.1.3. Algorithm:-

From the algorithms mentioned the wall follower algorithm was chosen. The wall follower algorithm uses either left or right-hand wall follower. Unlike the Flood Fill, the wall follower can be easily implemented and offers a guaranteed solution of the maze. Figure 28 shows the flow chart describing how the algorithm functions. It starts by scanning the area using the sensor, collects the incoming data by using code mentioned in section 5.1.2 and compares it with set of statements mentioned below:

1. If you can turn left, then turn left
2. Else if you can't turn left , then move forward
3. Else if you can't move forward, then turn right
4. Else turn around

If one of the conditions is true then the robot executes a movement and prints out the movement of the robot by writing symbol 'X' at current cell and inserts an arrow symbol in a new cell to indicate where the robot is facing after the movement. Finally, it prints out the updated map of the maze. The program repeats the code until the robot exits the maze.

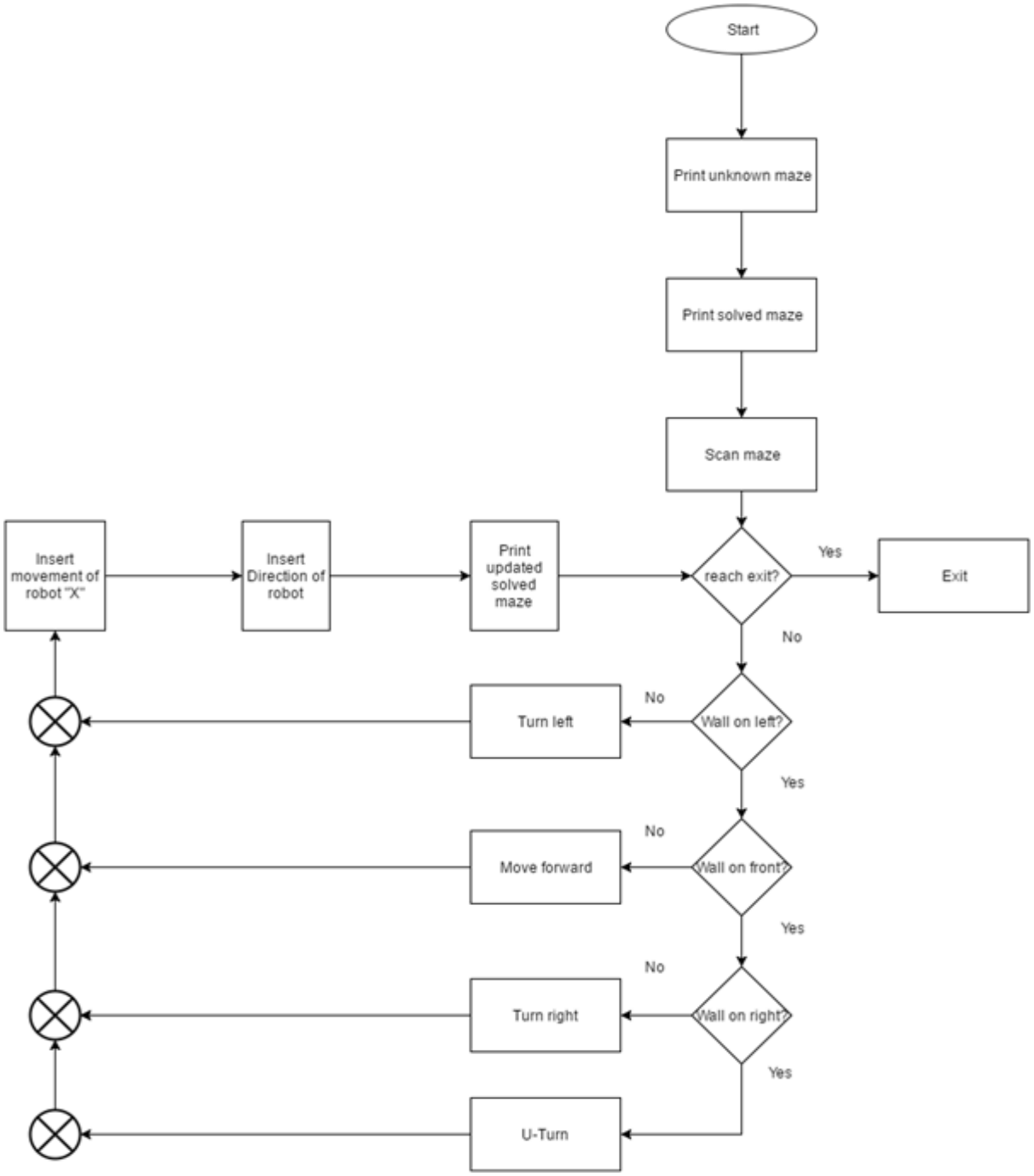


Figure 28: Left-hand wall follower flowchart

5.2. Robot Design:

5.2.1. Main Robot Block Diagram:-

As shown in Figure 29, it represents the block diagram of the main robot where it displays the connections made for the robot. A 9V battery is connected to the Arduino board as the voltage regulator regulates it to 5V which is a safe operating voltage for the Arduino. The 5V is then fed to power PING))) sensor, Servomotor and OpenLog and 3.3V to power HC-05 Bluetooth module. Another 12V power supply is connected to the motor controller chip

ULN2803 and motors. The PING))) sensor is used to detect the maze walls and transmit their location back to the Arduino. The PING))) sensor is placed in mounting bracket that is connected to servo motor where it allows the robot to scan maze from angle of 0^0 to 180^0. OpenLog is used to store the data coming from the microcontroller. The HC-05 Bluetooth module is configured to be Master and is used to transmit the data to the follower robot.

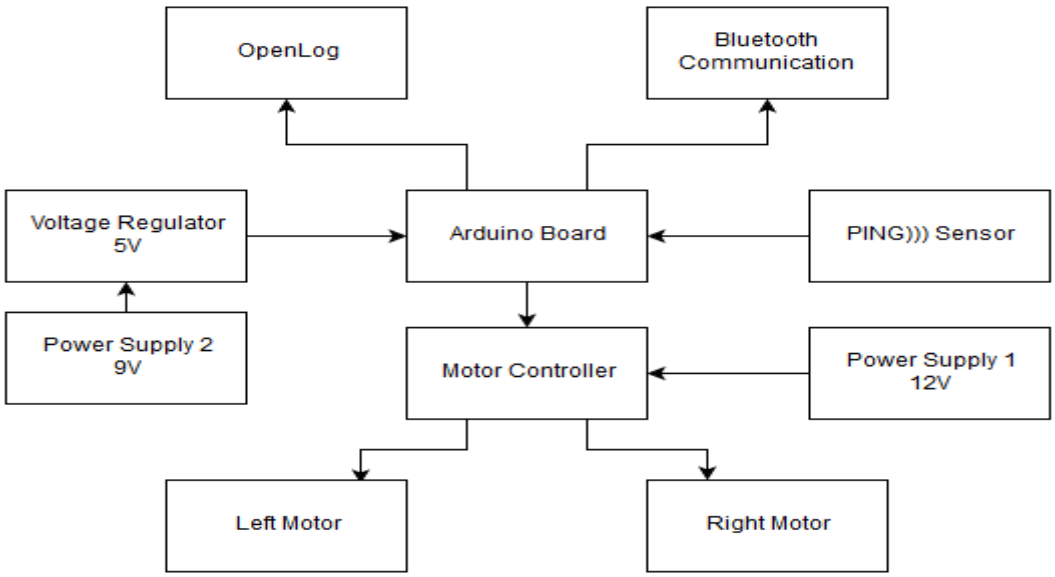


Figure 29: Main robot block diagram

5.2.2. Follower Robot Block Diagram:-

Figure 30 represents a block diagram of the follower robot; it is similar to figure 27 except in that it does not contain a sensor or data logger. In the follower robot, the HC-05 Bluetooth module is configured to be Slave and is used to receive the data coming from the main robot.

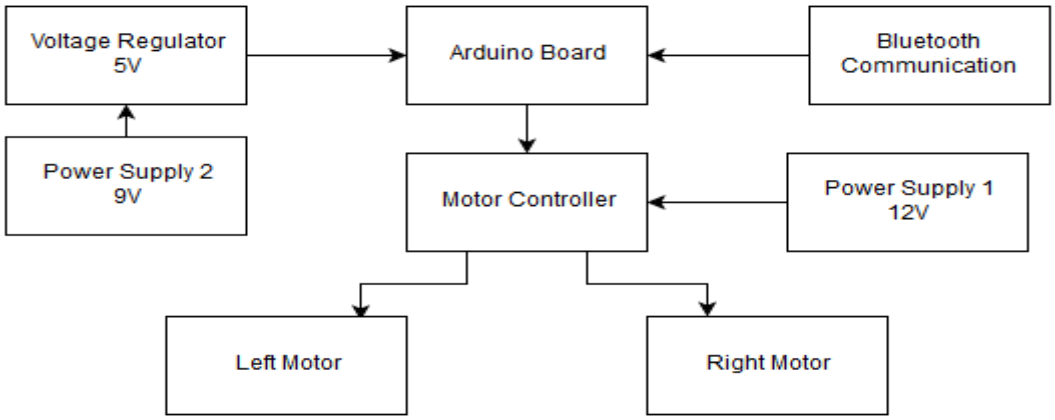


Figure 30: Follower robot block diagram

5.2.3. Motor Drive Circuit:-

Figure 31 represents the Darlington transistor array chip used in motor drive circuit and Figure 32 displays the motor controller circuit with Figure 33 to display the connections. The chip used is a ULN 2803 which is a Darlington transistor array that contains 8 NPN Darlington pair which is used for switching loads. Using the equation of ohm's law the value of resistance needed to limit the current going to the motor is calculated.

$$V = I * R$$

$$R = \frac{V}{I}$$

$$= \frac{12}{0.1}$$

$$= 120\Omega$$

Moreover, the battery used is Li-Po battery with a minimum capacity of 1000 mAh and constant discharge of 20C. Figure 34 displays the Li-Po battery used in the circuit. As the motor and chip draw continuously around 500mA each so we need to calculate the robot runtime dictated by battery capacity.

$$Battery\ Runtime = \frac{Battery\ capacity}{Current\ drawn}$$

$$= \frac{1000\ mA\text{h}\ capacity}{1000mA\ drawn}$$

$$= 1\ \text{hr}s = 60\ mins$$

Figure 35 shows a low voltage alarm used in the robot. This alarm is used to notify the operator that the Li-Po battery is about to reach 3.3V, operation below which could damage the battery.

Figure 31: ULN 2803 Transistor

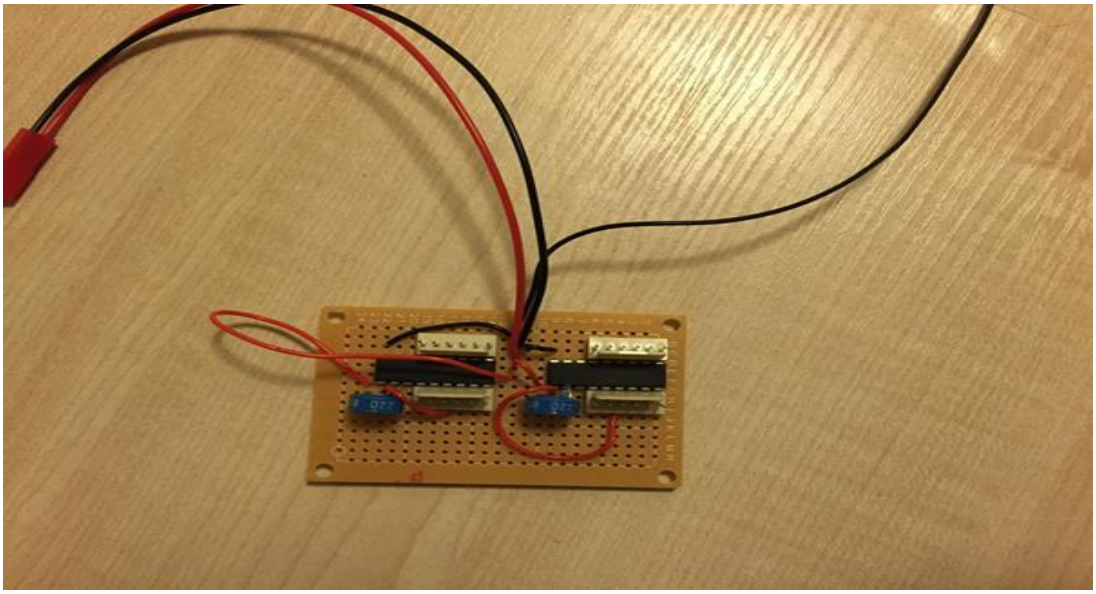


Figure 32: Motor controller circuit

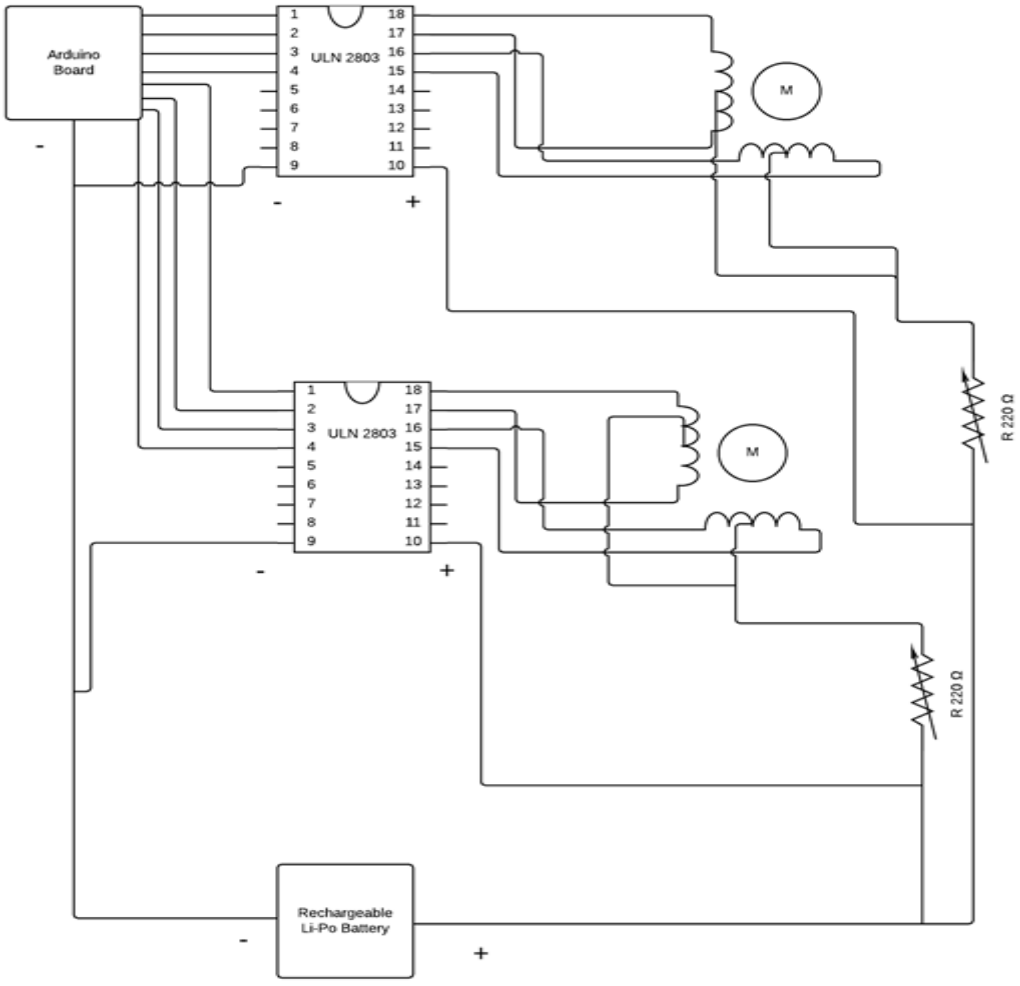


Figure 33: Motor controller circuit schematic diagram

Figure 34: Li-Po battery

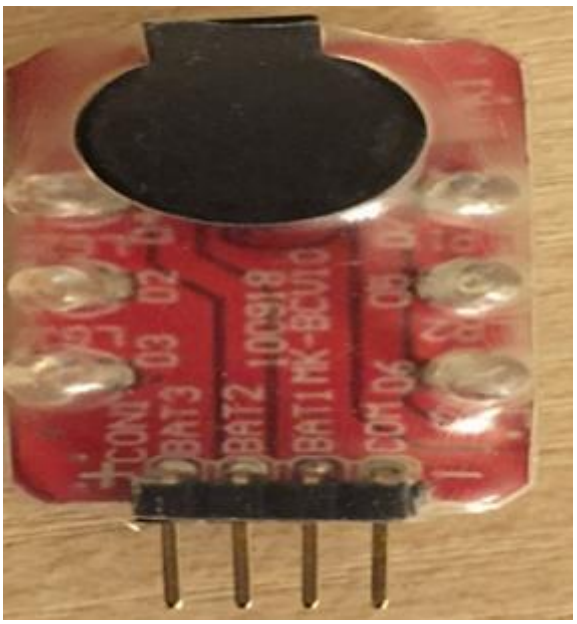


Figure 35: Low voltage alarm

5.2.4. Robot Fritzing Diagrams:-

Figure 36 and Figure 37 show the circuit configuration of the main and follower robot circuits. Figure 38 and Figure 39 show photographs of the final design of the main and follower robots. In the main robot, the PING))) sensor and the servo motor are placed at the top of the robot chassis. A box is placed at the bottom of the chassis which contains the rechargeable 9V battery, Li-Po battery, and low voltage alarm. On top of the box is the Arduino MEGA and surrounding the box is OpenLog, HC-05 Bluetooth module, and motor drive circuit.

The Follower robot possesses the same configuration with the PING))) sensor, OpenLog and servo motor removed.

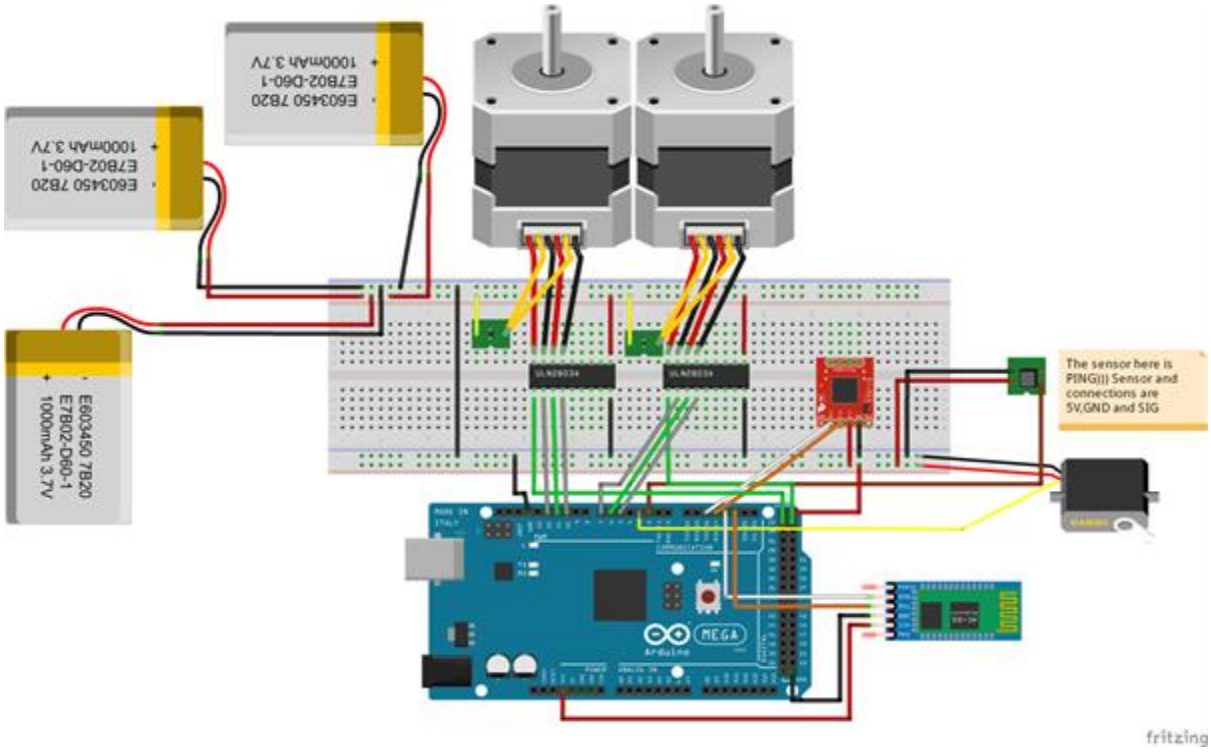


Figure 36: Main robot configuration

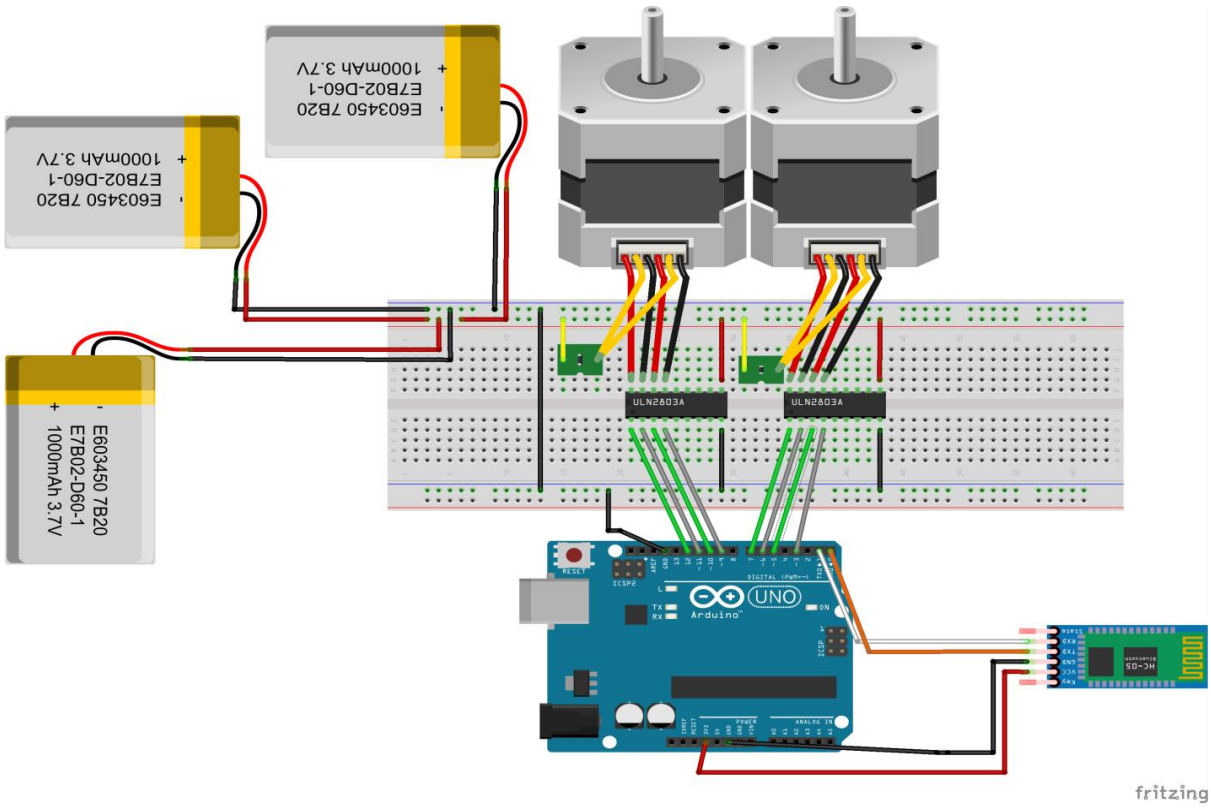


Figure 37: Follower robot configuration

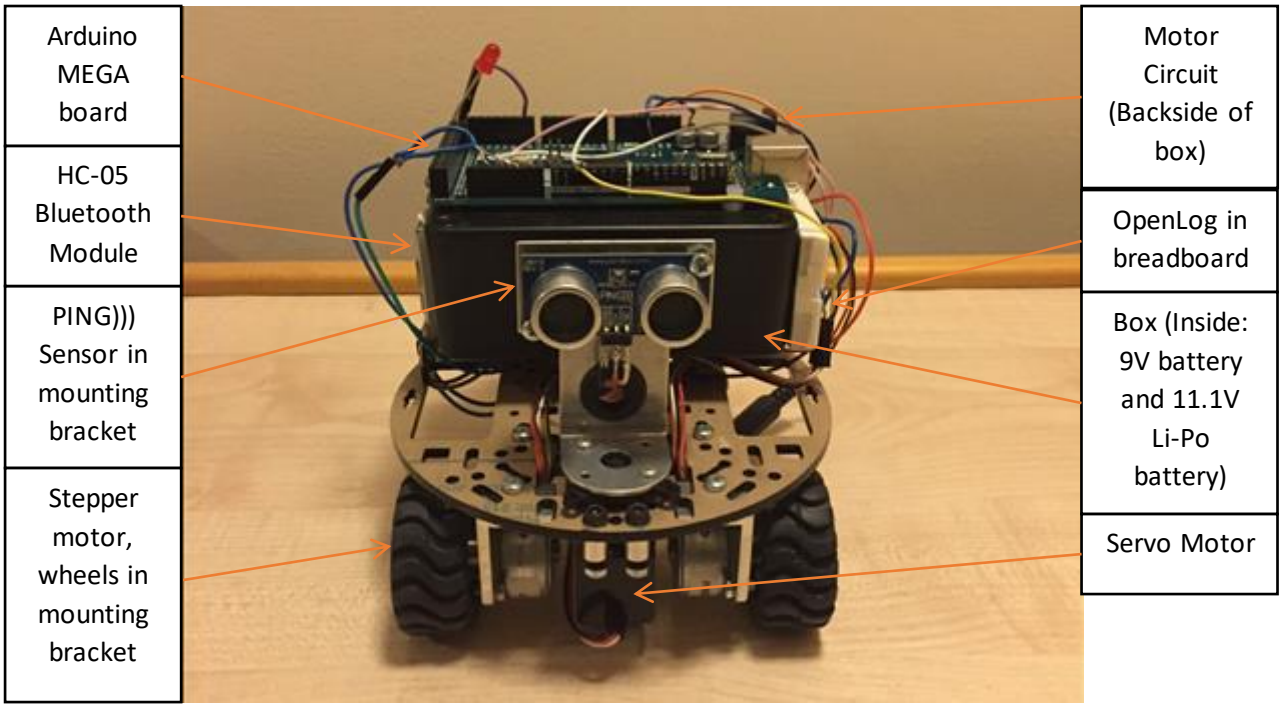


Figure 38: Final Design of main robot

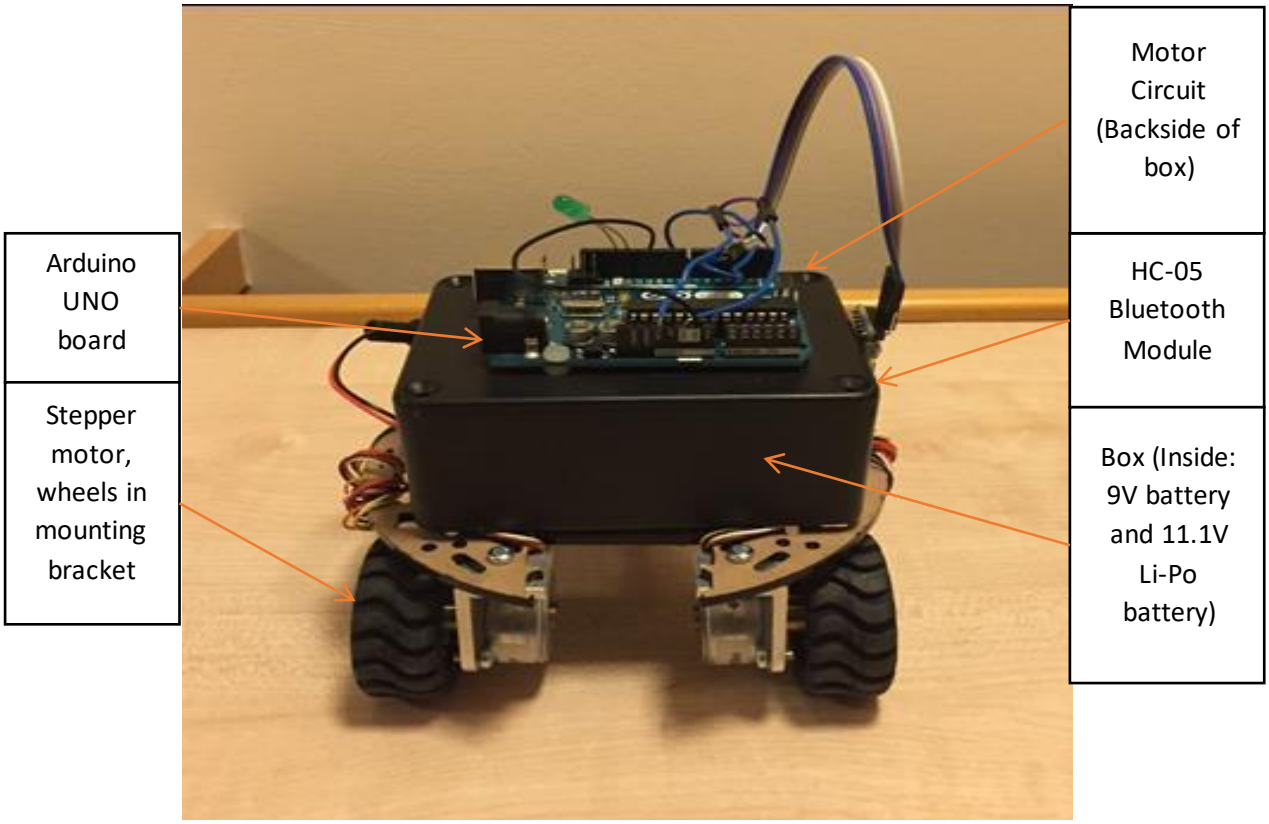


Figure 39: Final Design of follower robot

5.2.5. Robot Program:-

Two robot programs were developed and uploaded to the main and follower robots. The main robot's program consists of the left-hand wall follower algorithm, Bluetooth communication algorithm, map saving algorithm to the microSD card and movement algorithm. The second robots program contains Bluetooth communication algorithm and movement algorithm.

1. Main Robot Program:

Initially, the robot prints the maps of an unknown maze and solves it with an indication of the entrance and exit. After that, it transmits the data to the second robot and saves it in microSD card. The code is shown below.

```
void setup()
{
 pinMode(LEDPin, OUTPUT);        // Set the Pin to output
 servo.attach(servoPin);         // Set the Servo pin to output
 Serial.begin(9600);             // Setting the baud for serial monitor
 Serial1.begin(9600);            // Setting the baud for bluetooth
 Serial2.begin(9600);            // Setting the baud for OpenLog
 delay(1000);                    // Wait a second for OpenLog to initiate
 // Printing Unknown Maze map
 Serial.print("Unknown MAZE");
 Serial.print("\n");
 Serial1.write("Unknown MAZE");
 Serial1.write("\n");
 Serial2.write("Unknown MAZE");
 Serial2.write("\n");
 UnknownMaze[10][1] = 'E';       // Set the entrance of the robot
 UnknownMaze[0][9] = 'T';        // Set the exit of the robot
 UnknownMap();                   // Call the unknown map Function
 delay(2000);
 // Printing Solved Maze map
 Serial.print("Solved Maze");
```

```
  Serial.print("\n");
  Serial1.write("Solved Maze");
  Serial1.write("\n");
  Serial2.write("Solved Maze");
  Serial2.write("\n");
  SolvedMap();                // Call the solved map Function
}
```

Straightaway, the robot enters the loop statement where it does the scanning procedure as mentioned in section 5.1.2. After scanning the area surrounding the robot it compares the scan values with declared variable which is “MinimumSafeDistance” and enters the priority statements of left hand wall algorithm and checks which statement is true.
When a statement is true, it turns on the required motor/s to move the robot. It then prints the movement of the robot and transfers the movement and the updated map of the solved maze to the follower robot. The code below is an example of the following procedure.
Figure 40 represents the flowchart diagram of the main robot program.

```
if (leftscan > MinimumSafeDistance)
   {
     Direction = 'L';
     Serial.print("Direction: ");
     Serial.println(Direction);
     Left();
     Serial1.write(Direction);
     Forward();
     // Showing the Movement of the Robot:
     SolvedMaze[rowrobot][columnrobot] = 'X';
     SolvedMaze[rowrobot][columnrobot-1] = 'X';
     SolvedMaze[rowrobot][columnrobot-2] = 'X';
     columnrobot = columnrobot-3;
     SolvedMaze[rowrobot][columnrobot] = '<';
     // Printing the Movement:
     digitalWrite(LEDPin, HIGH);      // Turn LED on to indicate that data is being transmitted
```

```
  SolvedMap();                  // Call the solved map Function
  digitalWrite(LEDPin, LOW);    // Turn LED off to indicate that data is transmitted
  delay(3000);
  …. (Rest of the code)
}
```

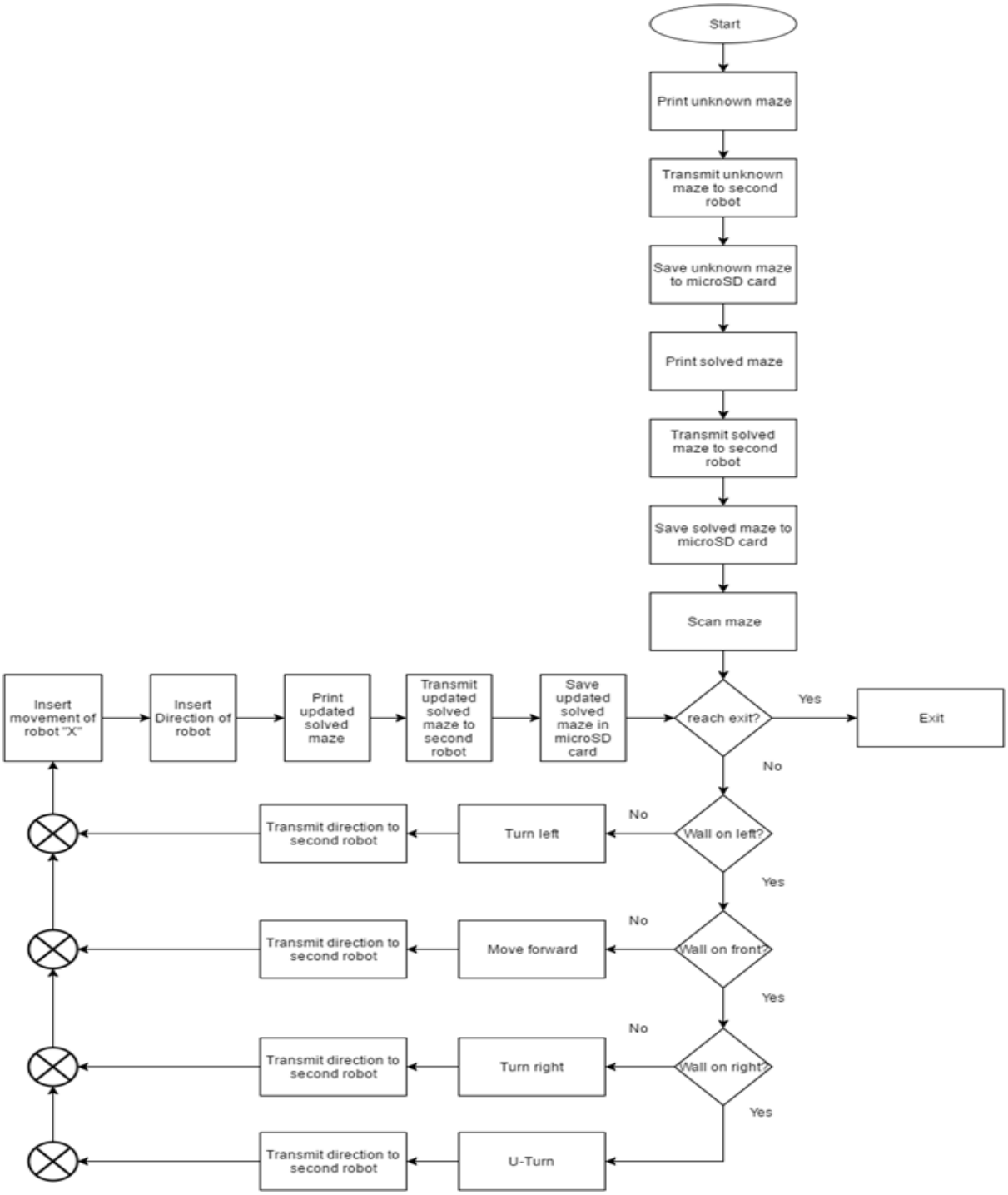


Figure 40: Main robot program

2. Follower Robot Program:

When the robot starts functioning, it receives the maps of the unknown and solved maze with entrance and exit in it. Later on, it waits to receive further incoming data that contains the direction that the main robot executed and compares it with a set of statements as shown below.

```
if(mySerial.available() != 0)
 {
   digitalWrite(LEDPin, HIGH);      // Turn LED on
   char inData = mySerial.read();
   Serial.print(inData);
   packet ++;
   digitalWrite(LEDPin, LOW);       // Turn LED off

   if (inData == 'L')
   {
    Left();
   }
   if (inData == 'F')
   {
    Forward();
   }
   if (inData == 'R')
   {
    Right();
   }
   if (inData == 'B')
   {
    Backward();
   }
}
```

If one of the conditions is true, the robot executes the movement algorithm to turn on the motor/s. After the movement, it receives the updated map of solved maze. Figure 41 represents the flowchart diagram of the follower robot program.

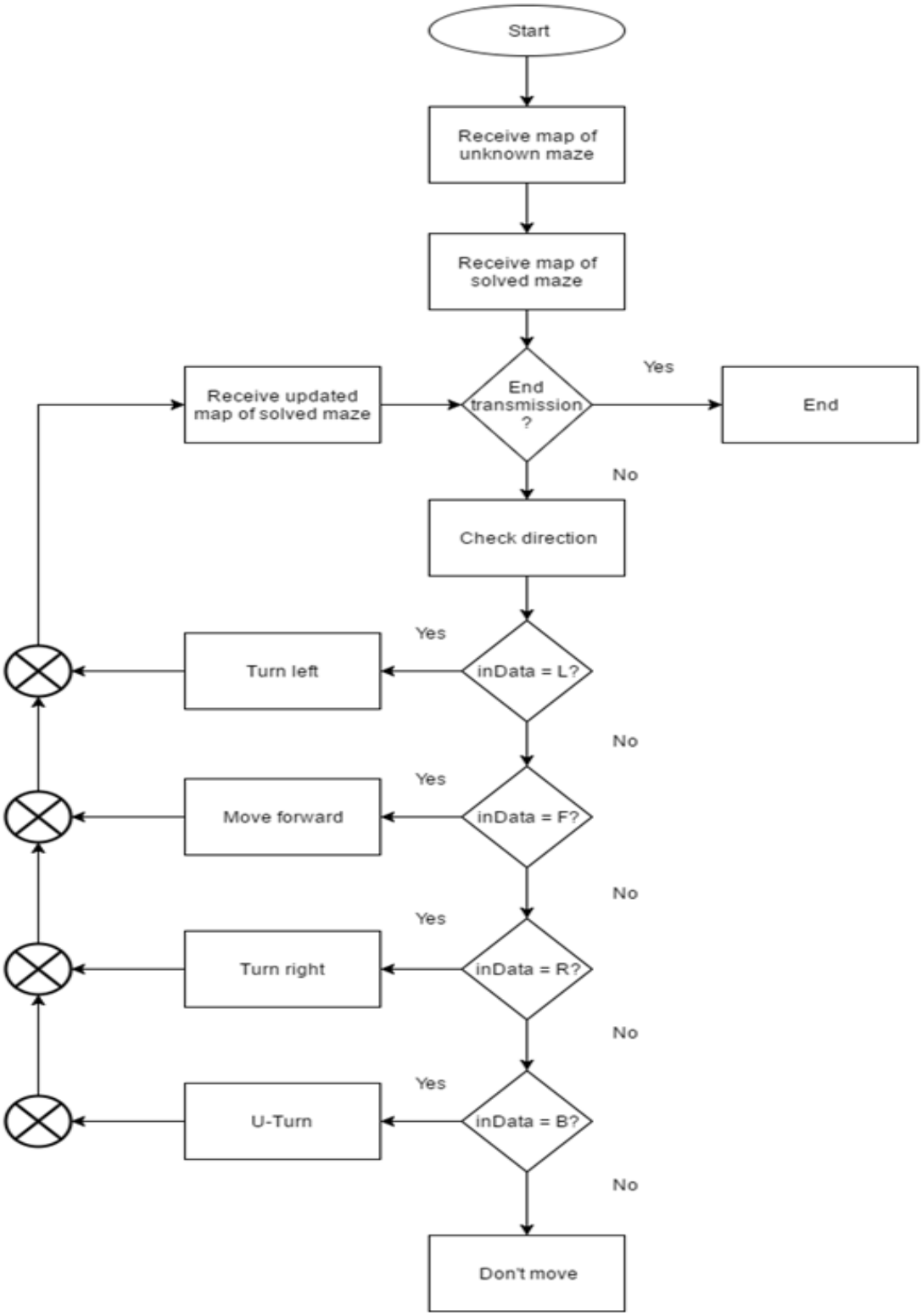


Figure 41: Follower robot program

5.3. Experimental Procedures:

A maze was constructed with 3x3 grid cells as shown in Figure 42. Initially, the main robot is placed inside the maze at the entrance as demonstrated in Figure 43 and it starts solving the maze. The microSD card is then removed from the robot and inserted into a computer to view the map stored in it. Lastly, the follower robot is placed inside the maze and follows the instructions received from the main robot. The results of the experiment are mentioned in next section.

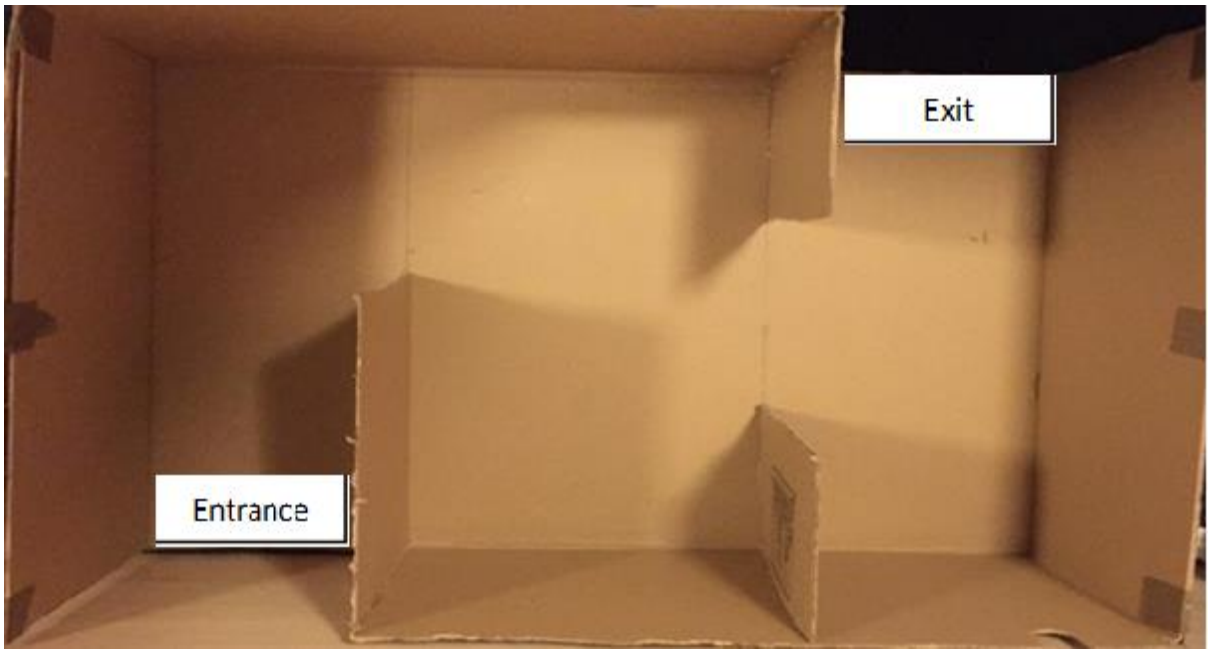


Figure 42: Maze structure

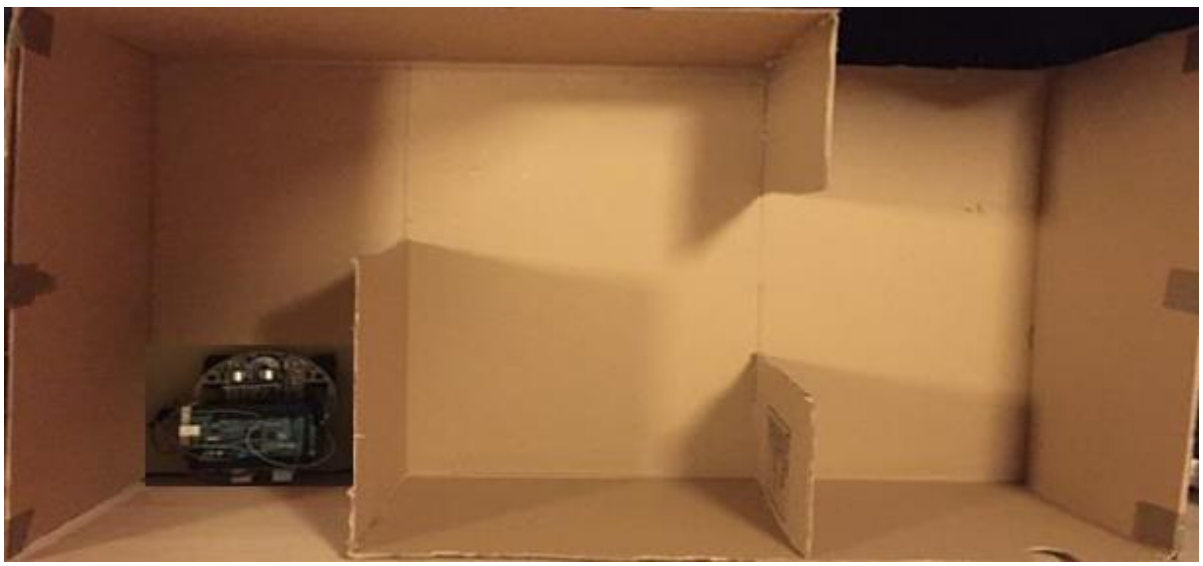


Figure 43: Robot inside the maze

6. Results:

As the program being uploaded to the robots, the main robot was able to solve the maze, store the map in microSD card as shown in figures 35 and 36. In figure 35, the map of the maze is printed in a form of a multidimensional array. The map marks “E” as the entrance of the maze, “T” as the exit of the maze, “X” to represent the movement of the robot and dashes represents the wall structures of the maze.

Figure 44:(a) Movement of the robot

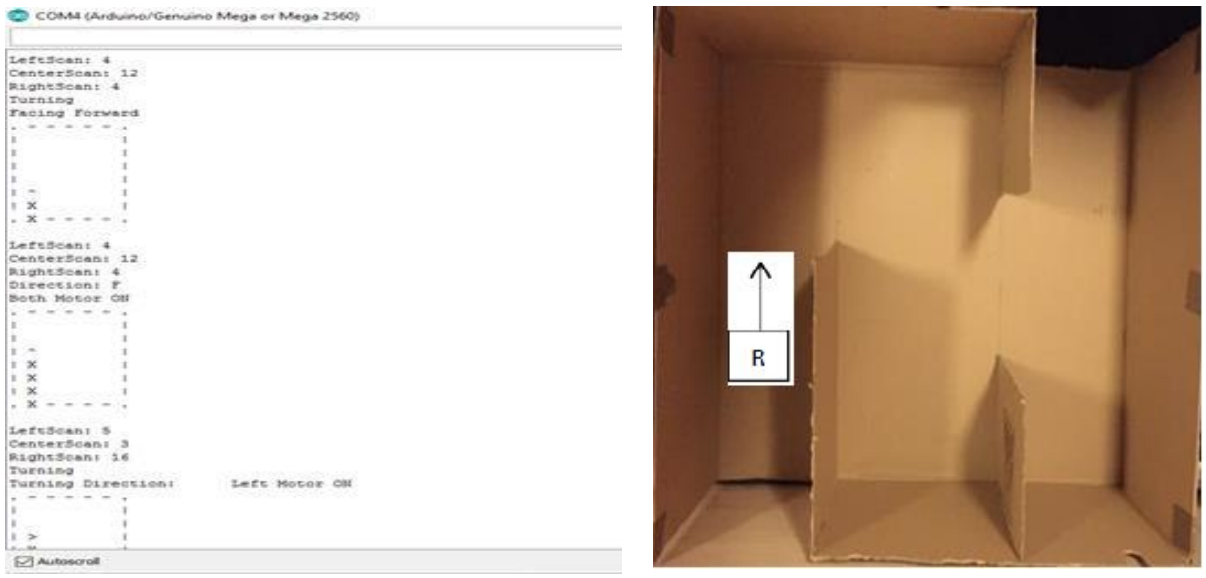


Figure 45:(b) Movement of the robot

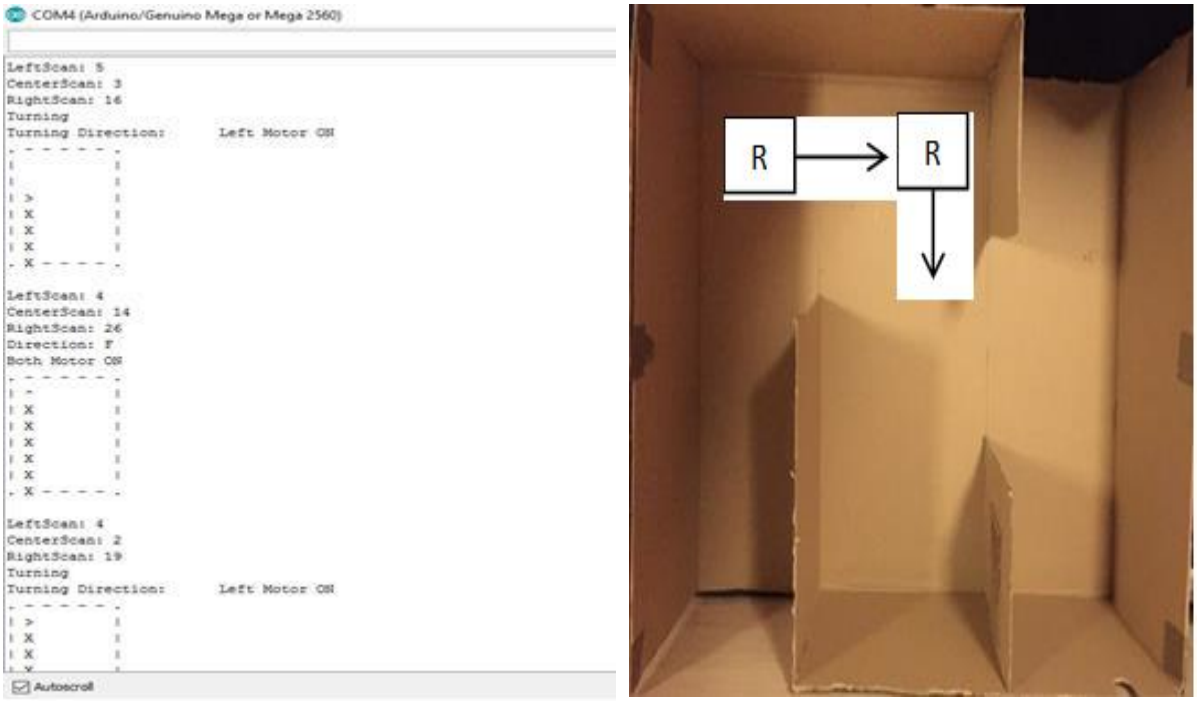


Figure 46:(c) Movement of the robot

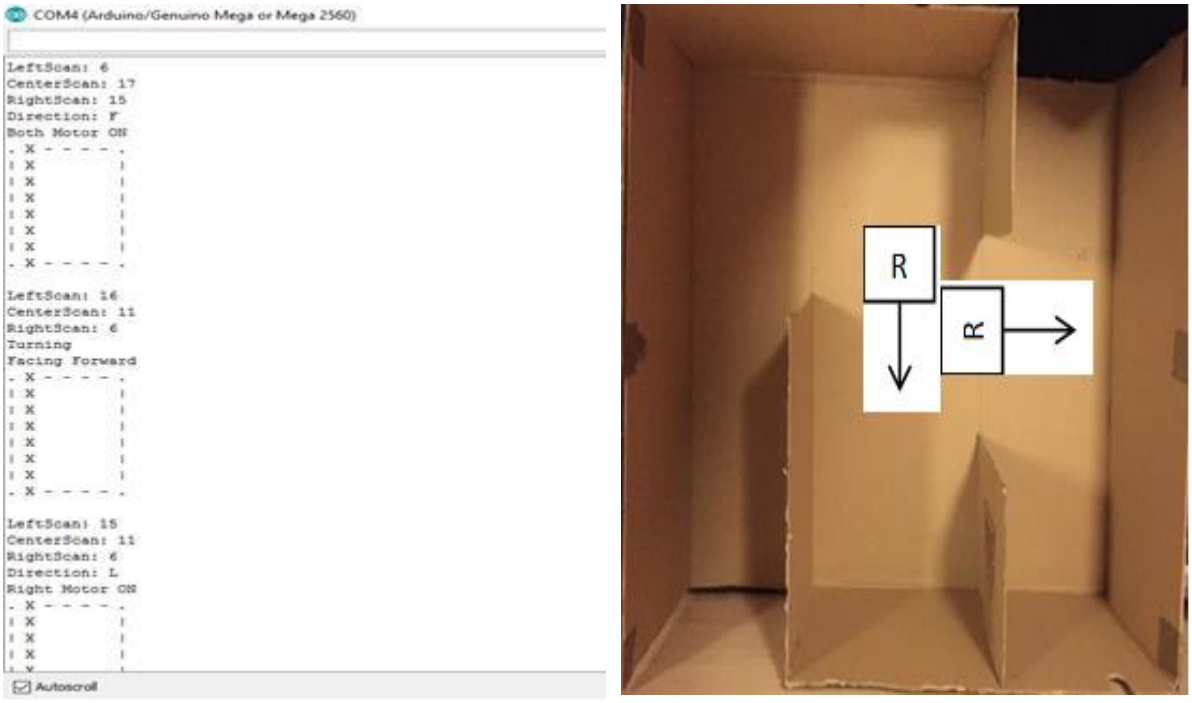


Figure 47:(d) Movement of the robot

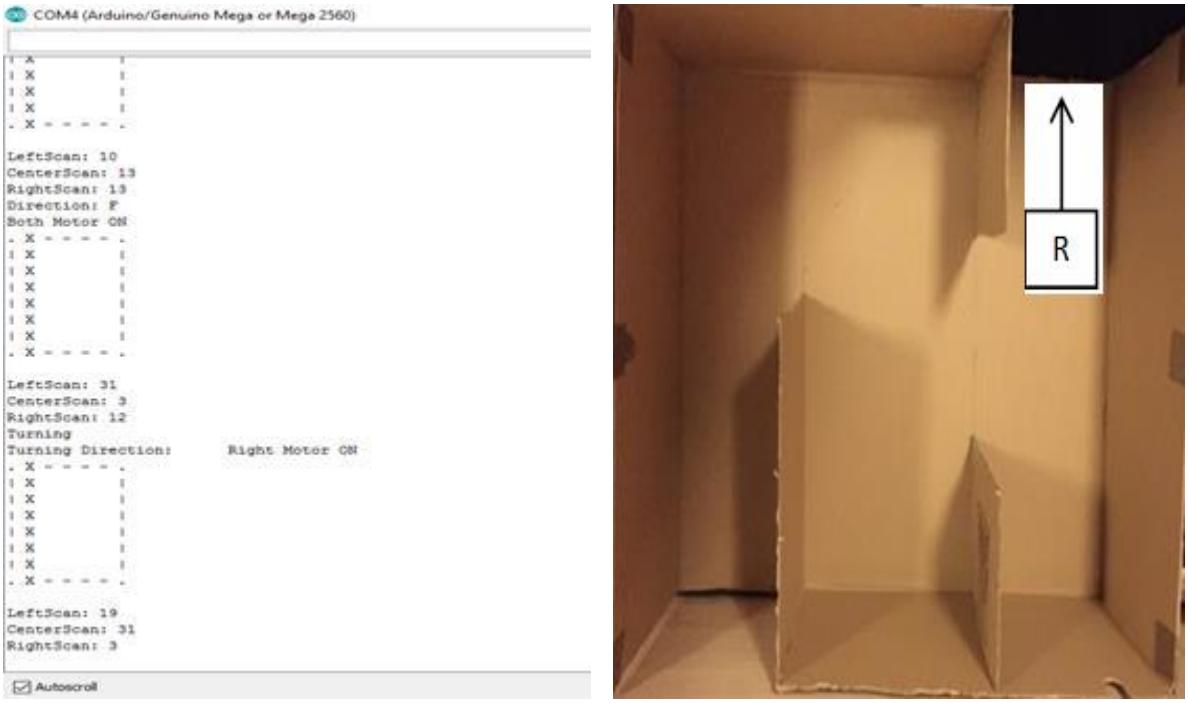


Figure 48:(e) Movement of the robot

As shown from Figure 44 to Figure 48, the robot was able to solve the maze but the movement of the robot is not consistent with the map of the maze. As presented in Figure 49 and Figure 50, the map of the maze is stored in microSD card and in a form of a text (.TXT) file.

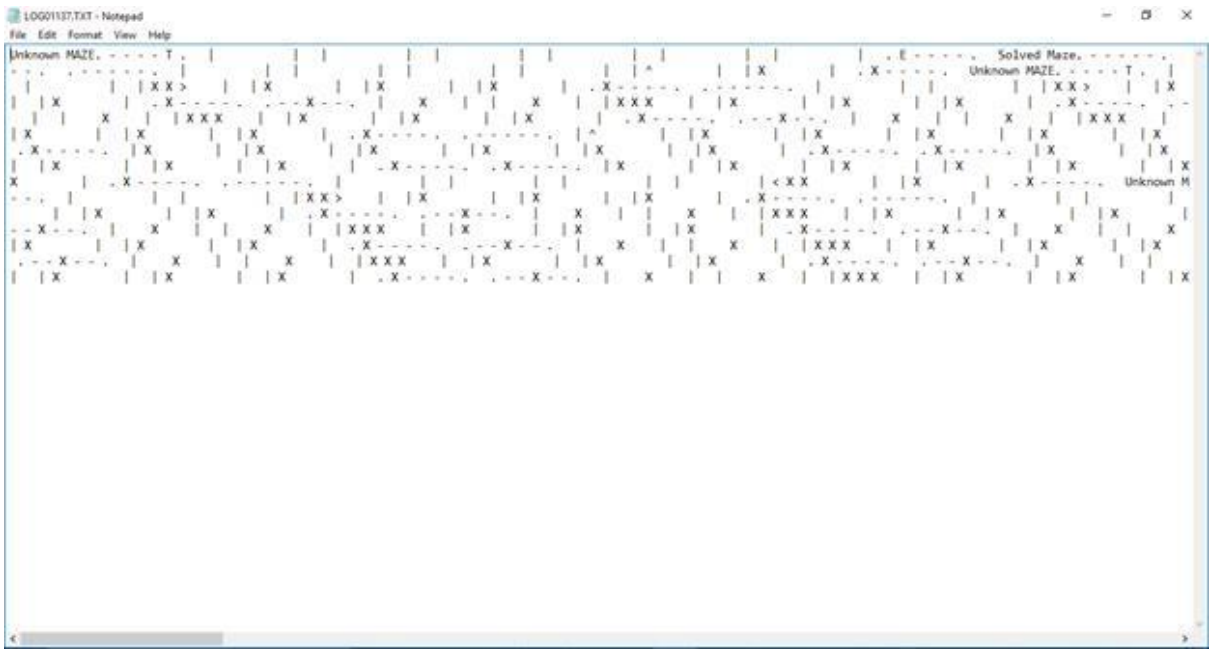


Figure 49: (a) Map of the maze in microSD card

Figure 50: (b) After rearranging Map of the maze in microSD card

While solving the maze, the main robot was able to send the map of the maze to follower robot as shown in Figure 51. Furthermore, the follower robot was able to receive the map of the maze and also receive the movements of the main robot where it was able to use these directions to solve the maze without the help of the sensors.

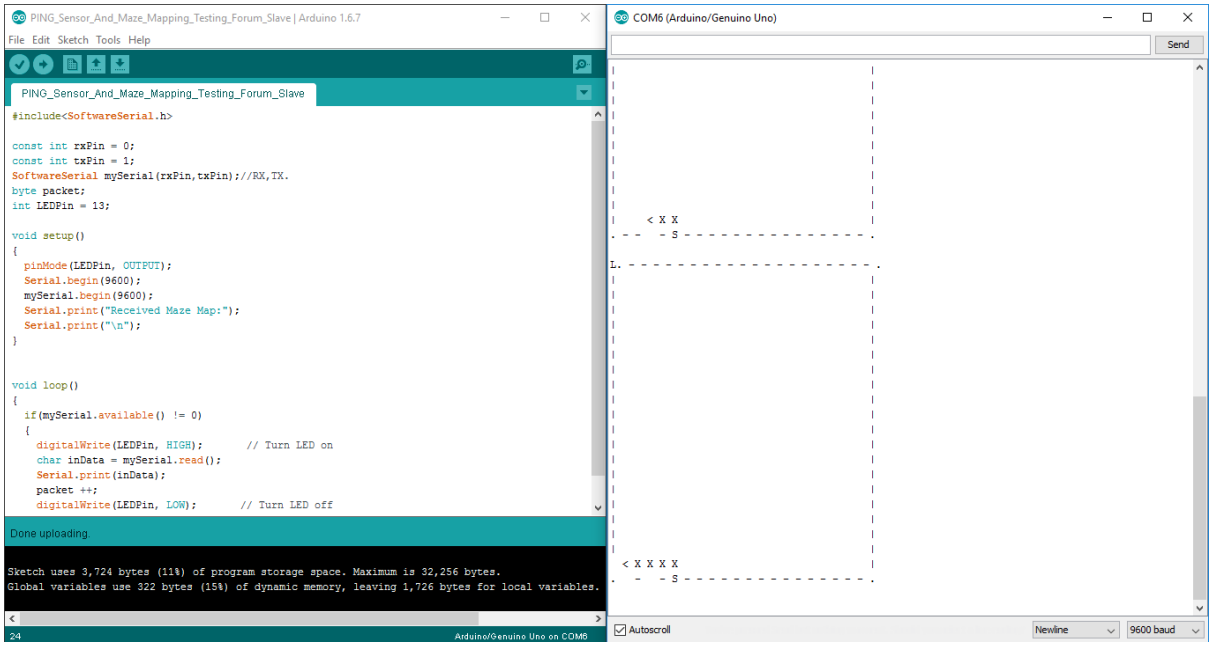


Figure 51: Transmission of the map of the maze to follower robot

7. Discussion, Conclusion and Further work:

7.1. Discussion:

During the testing phase, the hardware components were tested individually to understand their functionality and code for it before assembling the final design. For instance, the motor circuit was tested to know the safe operating voltage and current for the stepper motor, as a result, it can operate at a voltage of 11.1V and with the help of a variable resistor, we were able to limit the current to 0.12A (as excess current could overheat the motor and damage it). Figure 52 demonstrates the safety testing for stepper motor operation. The stepper motor was operating as expected, as ULN2803 Darlington array transistor was able to assign the PWM signals to different sections of the coil where it magnetizes the magnet to allow the rotation of the motor. In addition, the PING))) sensor was connected to the PWM port in the Arduino MEGA to test its response to an obstacle at a known distance from the sensor. This helped demonstrate the required operation before integration into the main robot, the testing performed is illustrated in Figure 53 and Figure 54.

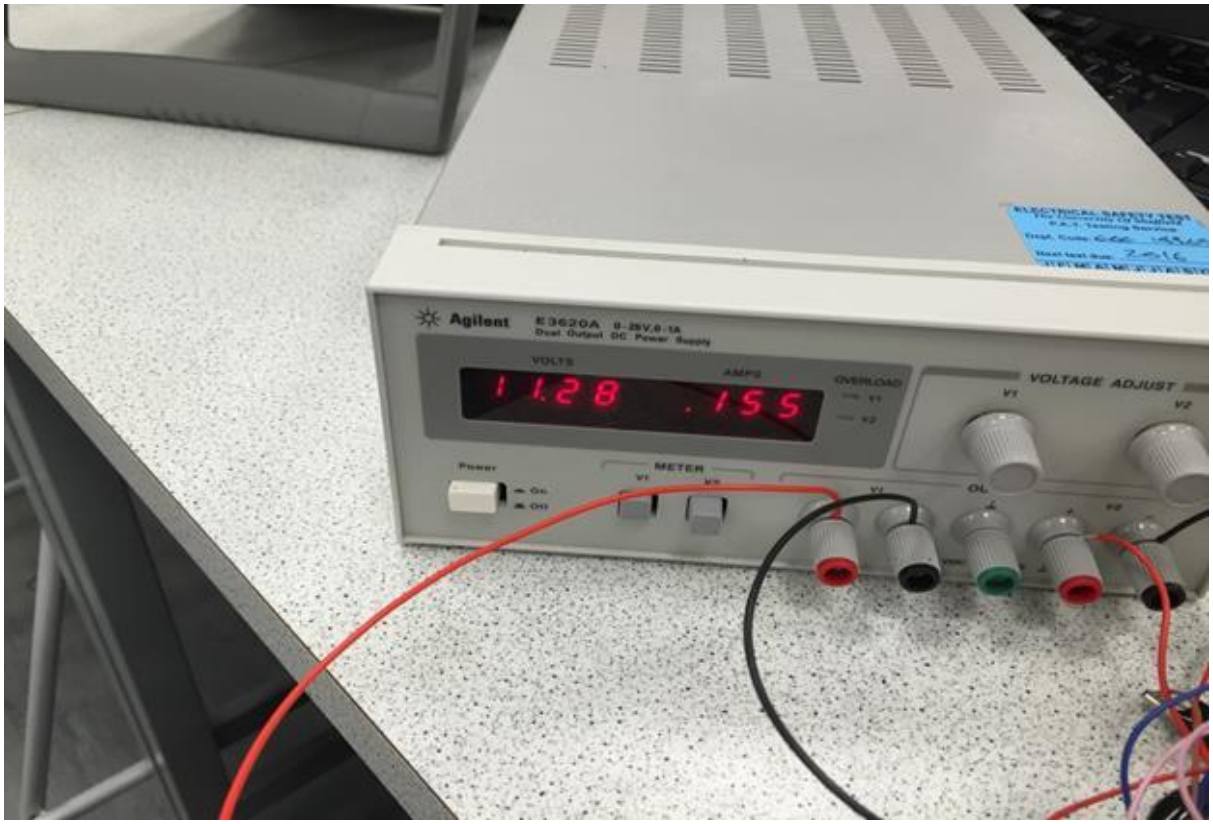


Figure 52: Safe operating voltage and current of the stepper motor

Figure 53: (a) Testing the PING))) sensor

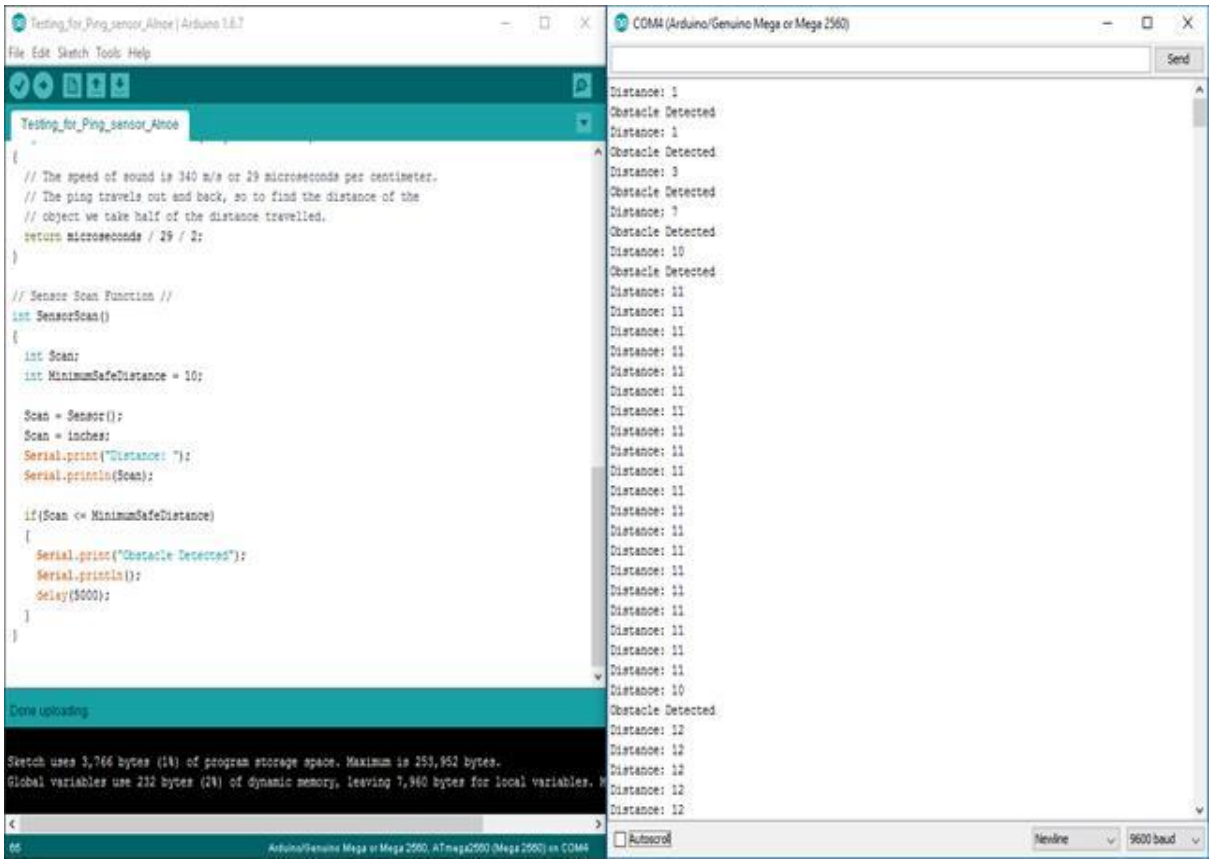


Figure 54: (b) Programming the sensor with result

After assembling the robots and testing the hardware individually, the code was combined and loaded to the microcontroller boards. The final testing was to see if the main robot was able to solve the maze, store the map and transmit it to follower robot by using the HC-05 Bluetooth modules. The experimental results are presented from Figure 44 to Figure 50. These results show how the main robot was able to solve the maze by scanning the area around it with the PING))) sensor using the left hand wall follower algorithm to decide which direction it should move next in addition the storage of the map is displayed in a form of text (.TXT) file however the map was stored horizontally this data can be rearranged to display the same result as illustrated in the serial monitor of Arduino IDE. In Figure 51, the result shown displays the map of the maze and the transmitted movements of the main robot in the follower robot. While testing both robots in the maze some issues were encountered; the motors of both robots were not able to provide enough force (torque) to move the robots, therefore, the testing was completed by pushing the robots around the maze. Moreover, the movement of the robots and map of the maze were disproportionate to each other. However, due to time constraints, no investigation was conducted in order to improve the mapping algorithm.

7.2. Conclusion:

Most of the goals of the project were achieved since both robots were able to navigate through the maze using the left-hand wall algorithm. The main robot was able to map the maze, store it in an external memory and transmits it to the follower robot. Additionally, the follower robot was able to receive the transmitted map and the movements of the robot to navigate through the maze. On the other hand, due to the difference between the mapping algorithm and the decision-making algorithm, the movement of the robot was inconsistent with the map of the maze. Throughout the testing, both robots were not functioning as expected due to a lack in the torque of the motors. This was tackled by pushing the robots through the course. The torque can be improved by adding gear/s.

7.3. Further work:

There were some challenges faced throughout the project in terms of hardware and software design. In hardware terms, both robots are not moving due to the problem in the

motor as the rated torque of the stepper motor is low. Table 10 and Table 11 symbolize the weight of both robots.

Table 10: Approximate weight of the main robot

Components	Weight/g	Quantity
Motor	31	2
Wheel	16	3
Body	40	1
Arduino MEGA	37	1
PING with bracket	136	1
Battery(11.1V)	87	1
Battery(9V)	45	1
Total Weight/g	392	
Total Weight/lbs	0.864212	
Diameter of Wheel/mm	42	
Diameter of Wheel/inches	1.65354	
Torque needed/oz.in	0.342962187	
Torque needed/g.cm	24.6959487	

Table 11: Approximate weight of the follower robot

Components	Weight/g	Quantity
Motor	31	2
Wheel	16	3
Body	40	1
Arduino UNO	25	1
Battery(11.1V)	87	1
Battery(9V)	45	1
Total Weight/g	244	
Total Weight/lbs	0.537928	
Diameter of Wheel/mm	42	
Diameter of Wheel/inches	1.65354	
Torque needed/oz.in	0.213476512	
Torque needed/g.cm	15.3719716	

Therefore, calculations were done to find out the required torque to push the robot and as a result from the calculations, it signifies that the torque needed is lower than the torque of motor but the rule of thumb is that the torque of the motor should be twice of the torque needed to move the robot. Equation used to calculate the Torque needed:

$$T = 8 * C * W * D$$

Where T is Torque of the motor, C is the Friction Coefficient, W is the weight of the robot and D is the diameter of the wheel. Therefore, to get at the bottom of this problem, would recommend:

1. Add gears:

   Adding gears would increase the torque but it would compromise the Revolutions Per Minute (RPM) of the stepper motor.

2. Buy motor with higher torque

   Moreover, in software term, based on the result of the map of the maze it has been shown that movement of the robot disproportionate to map of the maze but due to time constraint, the investigation was incomplete.

8. References:

[1] Arduino (2018) 'What is Arduino?' Arduino. Available at: https://www.arduino.cc/en/Guide/Introduction.

[2] Arduino (2022a) 'Arduino Mega Datasheet'. Arduino, pp. 1–18. Available at: https://docs.arduino.cc/static/dbd0a6eb970881b64825595f507fdce6/A000067-datasheet.pdf.

[3] Arduino (2022b) 'Arduino UNO Datasheet', pp. 1–13. Available at: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf.

[4] BakarSayutiSaman, A. and Abdramane, I. (2013) 'Solving a Reconfigurable Maze using Hybrid Wall Follower Algorithm', *International Journal of Computer Applications*. doi: 10.5120/14097-2114.

[5] Bhatt, A. (2011) 'Difference between Bluetooth and WiFi', *Engineers Garage*. Engineers Garage. Available at: https://www.engineersgarage.com/difference-between-bluetooth-and-wifi/.

[6] Electrical4U (2020) *Servomechanism | Theory and Working Principle of Servo Motor*, *Electrical4U*. Available at: https://www.electrical4u.com/servo-motor-servo-mechanism-theory-and-working-principle/.

[7] Gupta, B. and Sehgal, S. (2014) 'Survey on techniques used in Autonomous Maze Solving Robot', in *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*. doi: 10.1109/CONFLUENCE.2014.6949354.

[8] Helen (2019) *DC Motor vs Stepper Motor vs Servo Motor – Which Motor Should you Choose for Your Project?*, *Seeedstudio*. Available at: https://www.seeedstudio.com/blog/2019/04/01/choosing-the-right-motor-for-your-project-dc-vs-stepper-vs-servo-motors/ (Accessed: 10 May 2020).

[9] IEEE (2013) *IEEE R6 SWA Spring Meeting May 11-12, 2013 in San Diego CA*, *IEEE*. Available at: https://ieee-region6.org/2013/the-ieee-r6-swa-spring-meeting-scheduled-for-11-12-may-2013-in-san-diego-ca-at-ucsd/.

[10] iteadstudio (2010) 'HC-05 Bluetooth Module User's Manual V1.0'. iteadstudio, pp. 1–13. Available at: https://usermanual.wiki/Document/HC05UserManual.593762476/view.

[11] Joh, H., Yang, I. and Ryoo, I. (2016) 'The internet of everything based on energy efficient P2P transmission technology with Bluetooth low energy', *Peer-to-Peer Networking and Applications*. doi: 10.1007/s12083-015-0377-4.

[12] Kassim, A. M. *et al.* (2013) 'Performances study of distance measurement sensor with different object materials and properties', *Proceedings - 2013 IEEE 3rd International Conference on System Engineering and Technology, ICSET 2013*, pp. 281–284. doi:

10.1109/ICSEngT.2013.6650185.

[13] Micromouse USA (2016) *CAMM 2016 Competition Rules*. Micromouse USA. Available at: http://micromouseusa.com/wp-content/uploads/2016/04/CAMM2016Rules.pdf (Accessed: 1 May 2016).

[14] Mohammad, T. (2009) 'Using Ultrasonic and Infrared Sensors for Distance Measurement', *World Academy of Science, engineering and Technology*, 51, pp. 1–6. Available at: https://www.researchgate.net/publication/258883212_Using_Ultrasonic_and_Infrared_Sensors_for_Distance_Measurement.

[15] Nathan Seidle (2014) 'OpenLog Datasheet'. Github. Available at: https://github.com/sparkfun/OpenLog/wiki/Datasheet.

[16] Nawale, B. (2020) 'Free Online Circuit Simulator with Fritzing Software tool'. IoTDunia. Available at: https://iotdunia.com/fritzing-software-tool/.

[17] Parallax (2009) 'PING))) ™ Ultrasonic Distance Sensor (# 28015)', *Parallax*.

[18] Parallax (2013) 'PING ™ Ultrasonic Distance Sensor', *Parallax*.

[19] Sabin Mathew (2013) *Brushless DC Motor, How it works?*, *Lesics Engineers*. Available at: https://www.lesics.com/brushless-dc-motor.html.

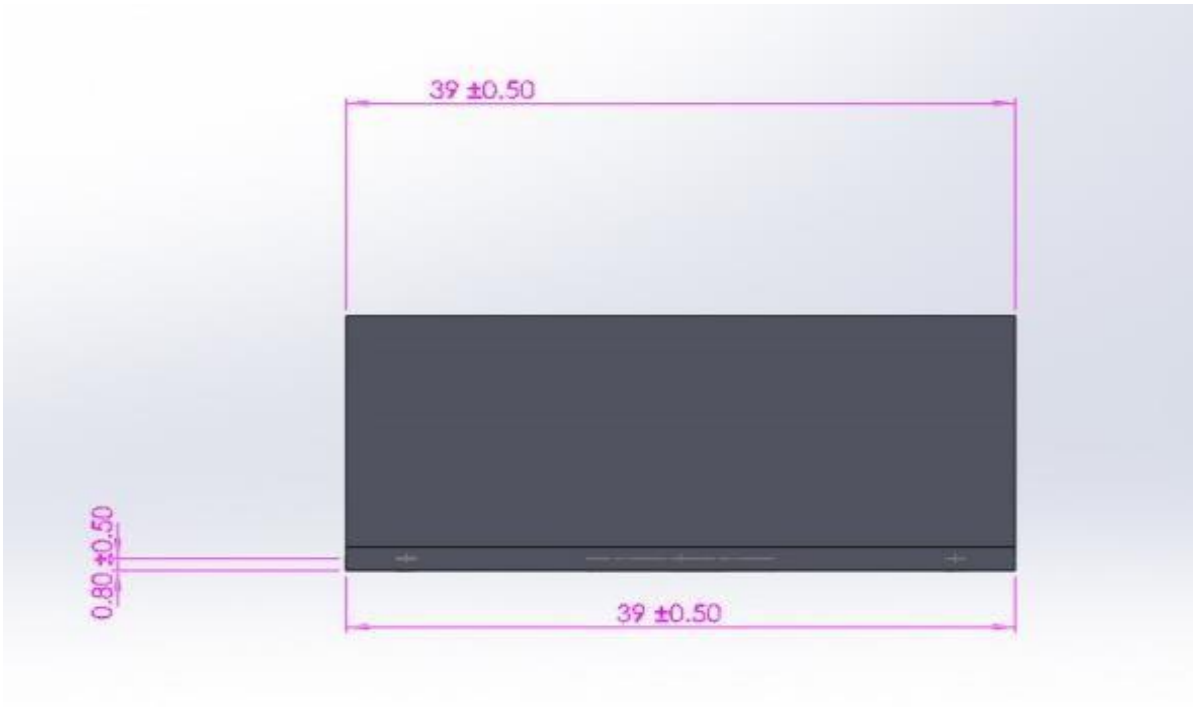
[20] Sabin Mathew (2014) *DC Motor, how it works ?*, *Lesics Engineers*. Available at: https://www.lesics.com/dc-motor-working.html.

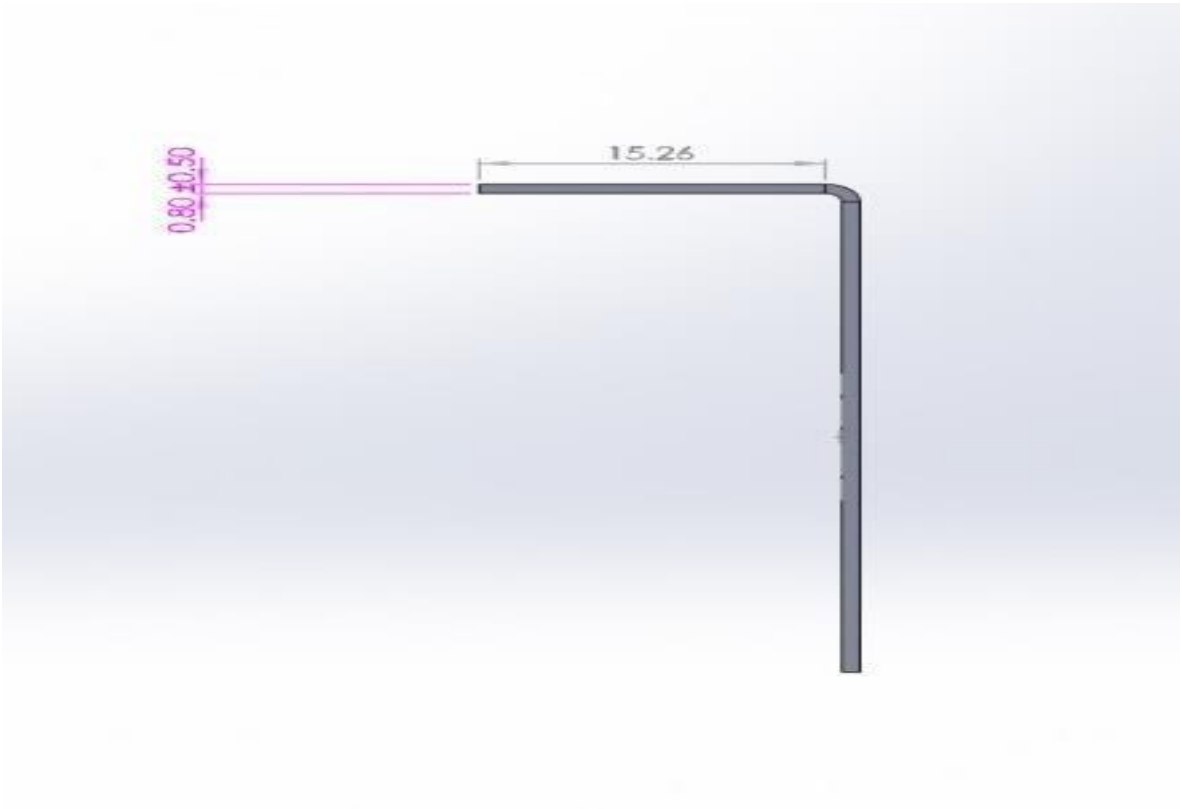
[21] Schweber, B. (2002) *Considerations in Choosing Motors for Robotics*, *Mouser Electronics*. Available at: https://www.mouser.co.uk/applications/considerations-choosing-advance-robotics/.

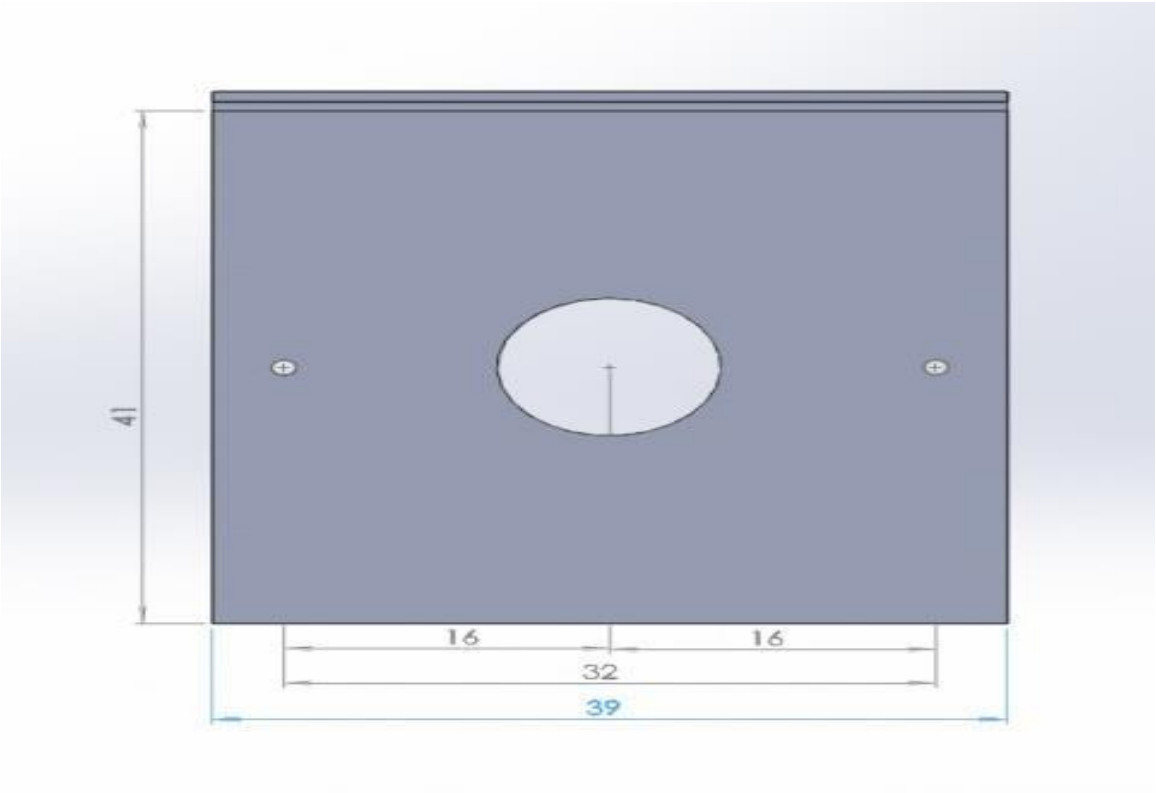
[22] SparkFun (2013) *Pulse Width Modulation*, *SparkFun Electronics*. Available at: https://learn.sparkfun.com/tutorials/pulse-width-modulation/duty-cycle.

9. Appendix:

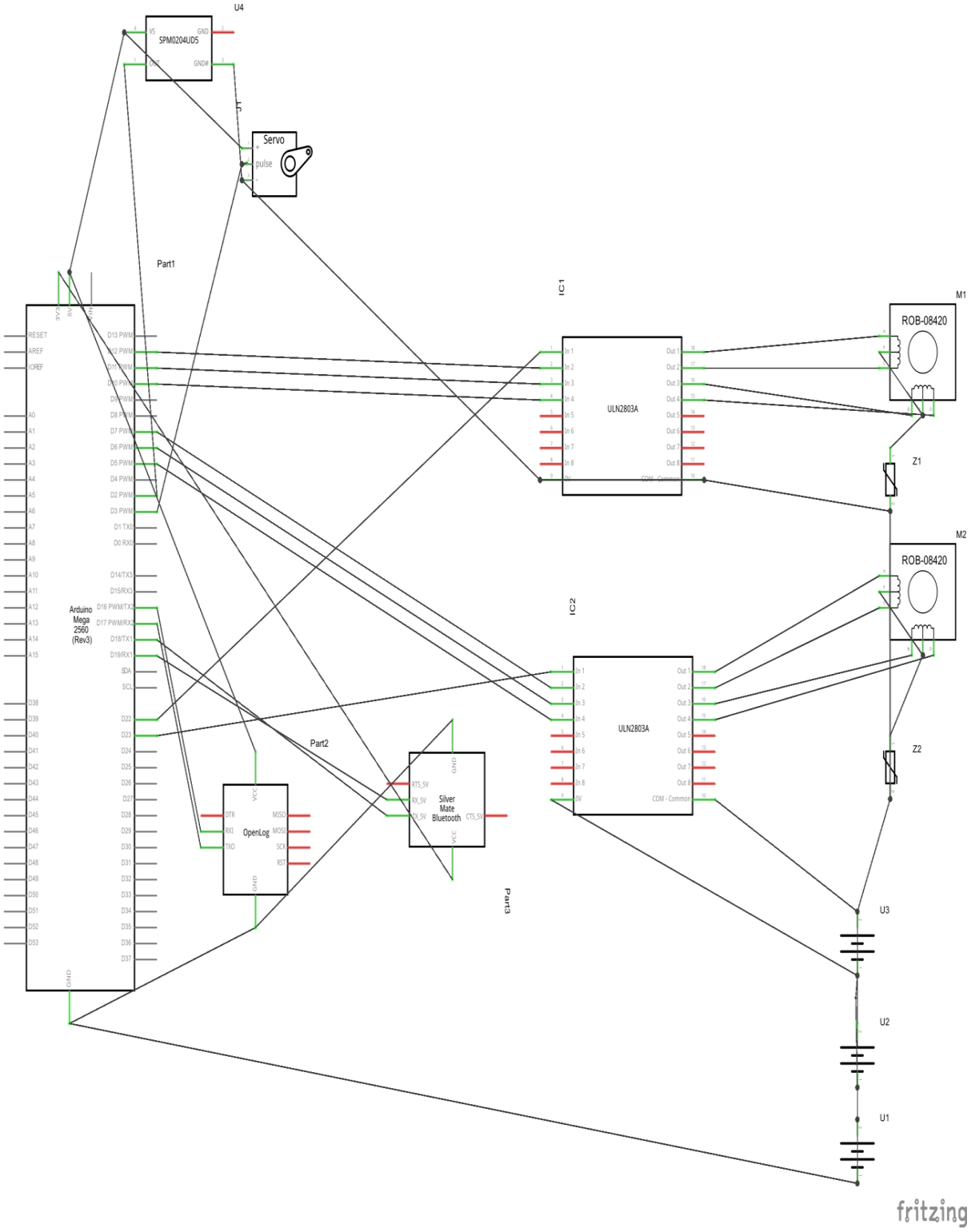
1) Solid-Work Design for the mounting bracket:






41
16
16
32
39

2) Main Robot Schematic Diagram:



3) Follower Robot Schematic Diagram: