SOUTH ASIAN INSTITUTE OF TECHNOLOGY AND MEDICINE (SAITM)

FACULTY OF ENGINEERING

DEPARTMENT OF MECHATRONICS ENGINEERING

ME 204-PRAXIS II

MAZE SOLVING ROBOT

FINAL REPORT

Group no : 05

Date of sub : 17. 11. 2017

Group member	ID number
A.N. Alexander	EN17003MEF0016
C.S. Perera	EN17003MEF0009
A.L.M. Siriwardhana	EN17003MEF0012

CONTENT

INTRODUCTION

Technology is developing rapidly and widely all over the world. So that introduction of new and automated technology is widely used for the day-to-day purposes of humans. The whole world can be automated in the future. Automated vehicles, arms, robots and some other machines are being introduced in now a days.

This project was done according to the above concept and the project is making an automated maze solving robot. The maze solving robot is a robot which solves a maze by identifying the junctions and obstacles automatically. These actions are done by motors, sensors and some other electronics devices to control the parts. The robot is consisted with two wheels connected to the motors and other 360^0 rotatable wheel. The 3 main stages of the robot is identifying the location, mapping and planning.

The whole process of the robot is depended on the identifying the location stage. Because robot can do other processes by identifying the location. The other stage is mapping. During identifying the location stage the mapping stage also followed by. Robot have to map the junctions and obstacles which are identified in the identifying location stage. After the mapping stage planning should be done to choose the shortest distance to complete the maze by avoiding obstacles.

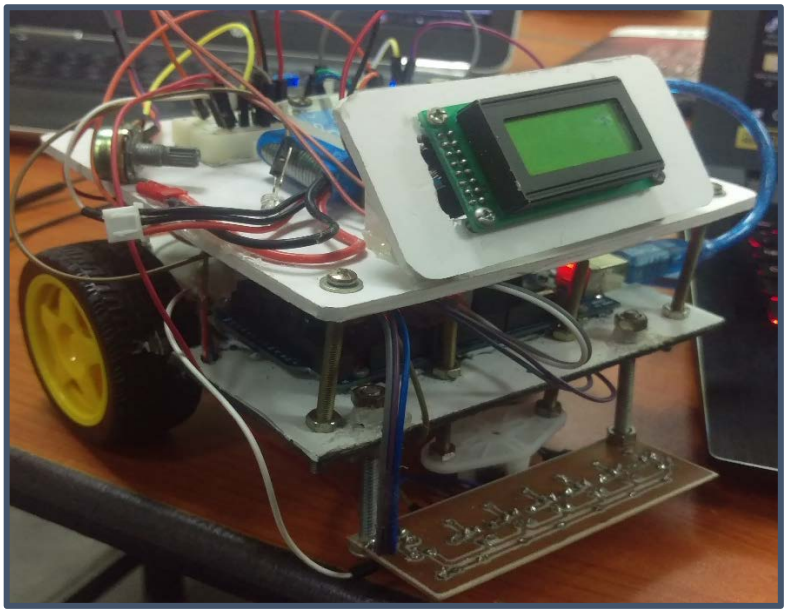


FIGURE: 01 picture of the robot

DESIGN SPECIFICATIONS

- **Parts of the robot;**
    - 2 motors
    - 2 wheels
    - A battery(7.5v)
    - A caster wheel (360^0 rotatable)
    - Arduino mega board
    - 6 TRCT5000 sensors
    - L298N motor driver
    - LCD display
    - Potentiometer
    - Wires
    - Boards (for the chassis)
    - Nuts and bolts
    - Resistors
    - Power supply

- **Chassis size** **:** 14cm*14cm*5cm
- **Full body size** **:** 16cm*14cm*14cm
- **Sensor array size** **:** 3.5cm*9cm
- **Wheel size** **:** 6.5cm (diameter)

The chassis of the robot is made using the 2 form boards and they are connected with bolts. The motors, wheels, Arduino board, motor driver and sensor array is connected to the lower board and battery, power supply and display is connected to the upper board. The LCD display, caster wheel and sensor array are in the front of the robot while other components are in the back of the robot.

- **The grid;**

The grid is on a white surface in black color. The lines of the grid is 3cm width and the grid is a 5*5 size square. There are obstacles and junctions on the grid randomly. The obstacles are in white color.

- **Assumptions**

    - The robot should travel the whole grid.
    - Robot should identify the junctions and obstacles separately.

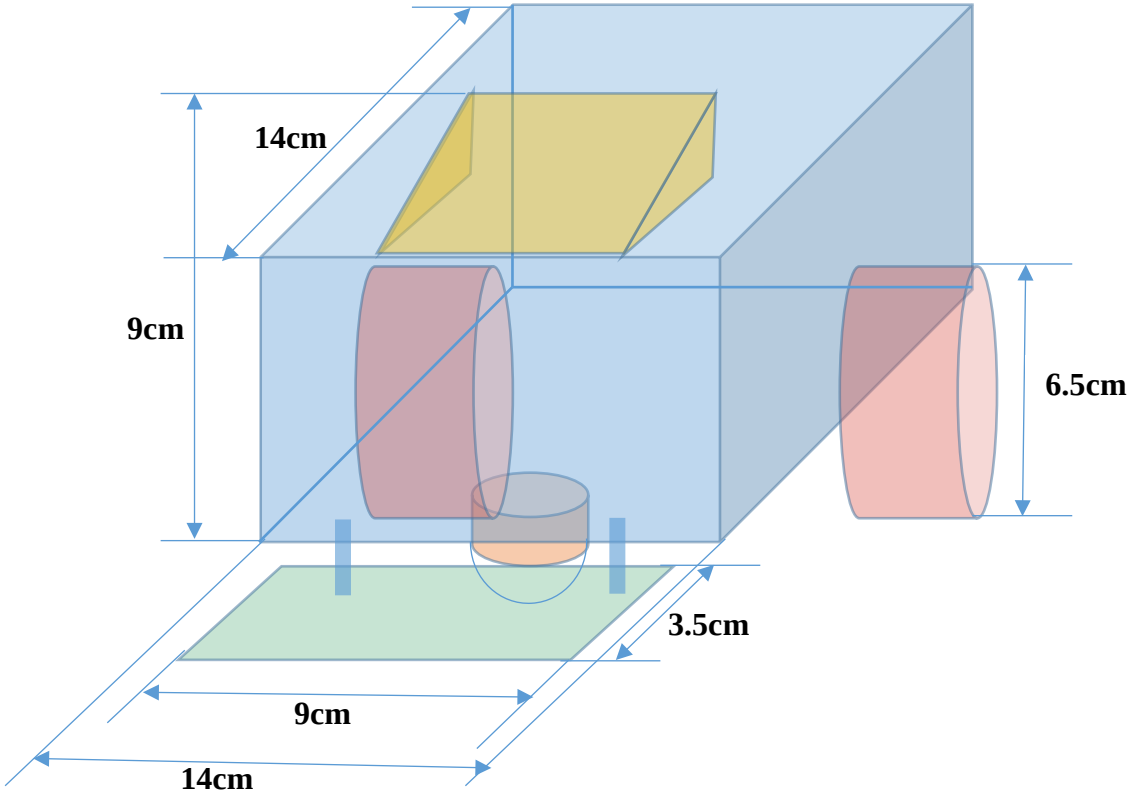


FIGURE: 02 dimensions of the robot

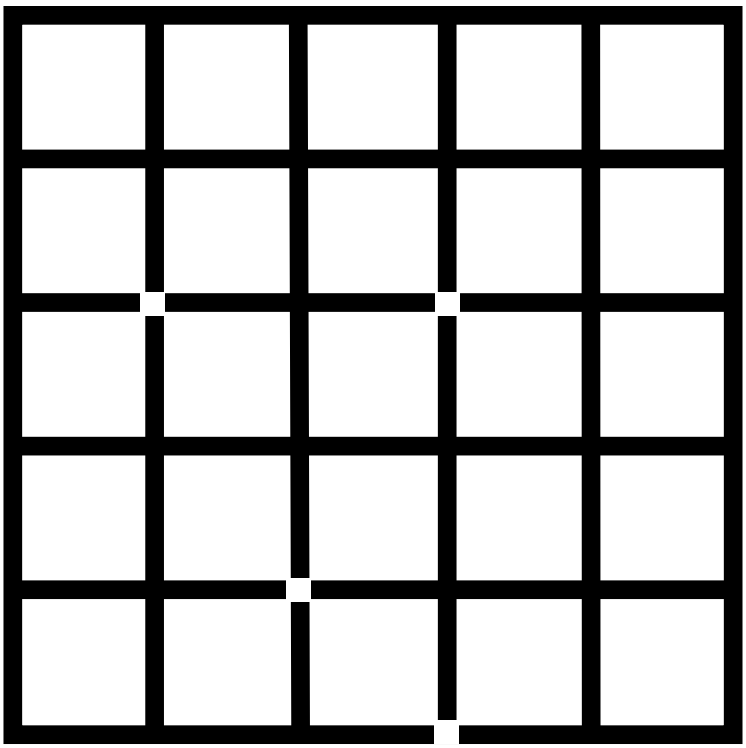


FIGURE: 03 sample grid

- **Arduino mega board**

Arduino board is the central processing unit of the robot. This board controls the all the electronic parts in the robot. This board is run by the Arduino software. The purpose of using an Arduino mega board is this robot wants high processing power. That power can be given by the mega board. The other reason is this board has many pins which is very useful for the robot. Because there are many pins to connect.

- **Motor and wheel**

Motor and wheel is the motion part of the robot. Here motor and wheel are considered as one component. Because the wheel is rotated by the motor. The motor is worked when low and high power is given to the two terminals of the motor. If both terminals are high or low the motor doesn't work.

- **L298N motor driver**

This is the controller of the motors. The terminals of the motors are connected to the motor driver. The speed of the motors are also controlled by this motor driver. It is called PWM. The speed can be controlled by giving a PWM value. The pins which are used to rotate or stop the motor are also located in this driver.

- **TRCT5000 sensors**

TRCT5000 is the sensor which is used for the line following or location identification stage. There are two bulb type Components in this sensor they are consist of a diode and a transistor. This sensor can identify black and white. This happens because of the IR rays. When transistor emit IR rays the diode detect the reflection of the emitted ray. Therefore this sensor is very useful for the line detection.

- **7.5V lithium battery**

7.5V Lithium battery is the power source of the robot. This is a rechargeable battery and gives a DC current to the robot all the electronic parts are powered by this battery.

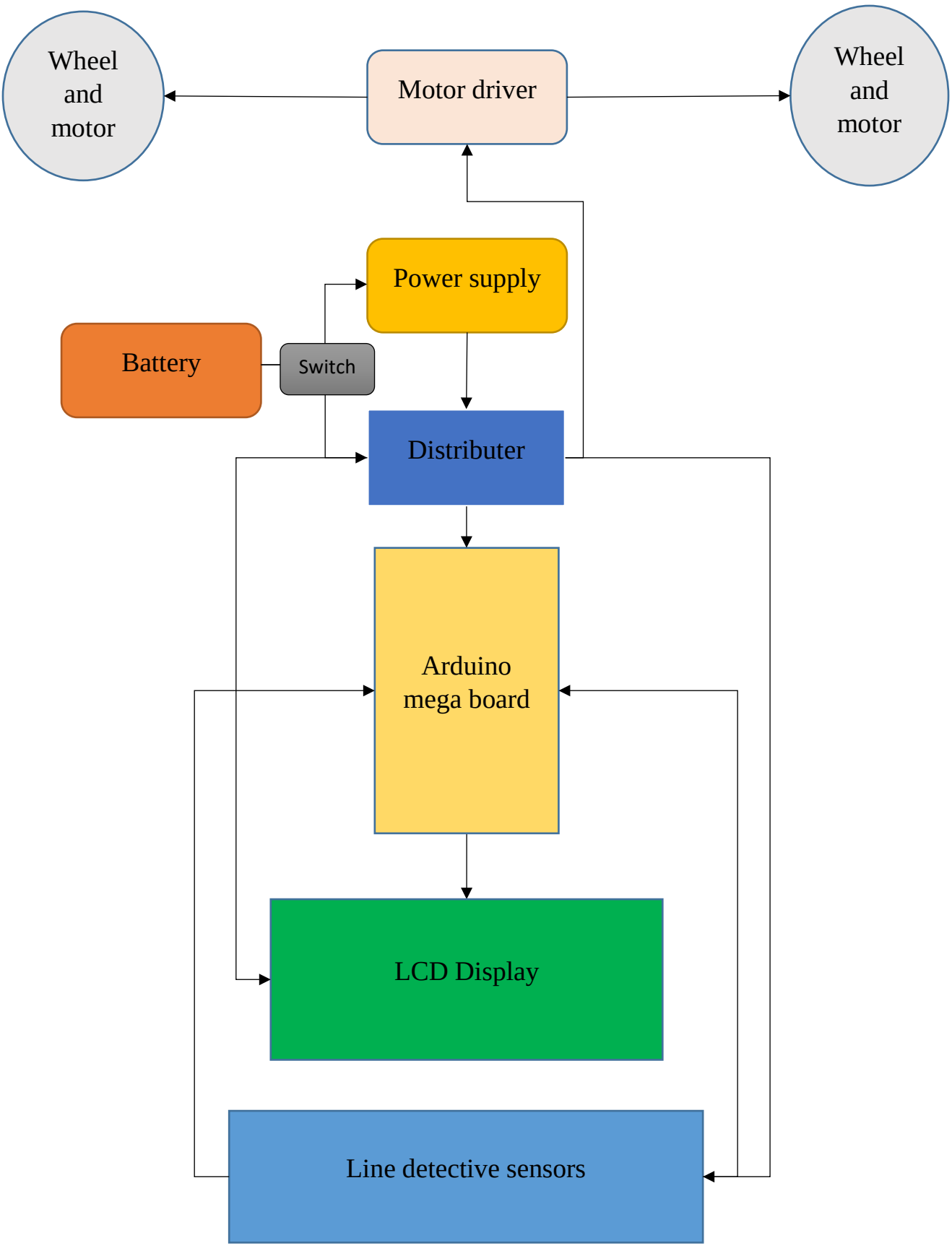
- **LCD display**

LCD display is the component that shows us the path, mode, directions etc. The display is 8 byte display and has upper and lower displays. This is also powered by the battery and Arduino.

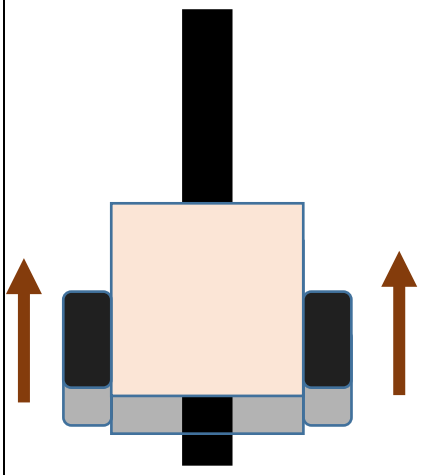
- **Power supply**

Power supply is the unit that converts the voltage in to different voltage state. Here 7.5V is converted in to 5V that is used for the LCD display, Arduino board and sensor panel. 5V is the voltage that should be given for the above components.

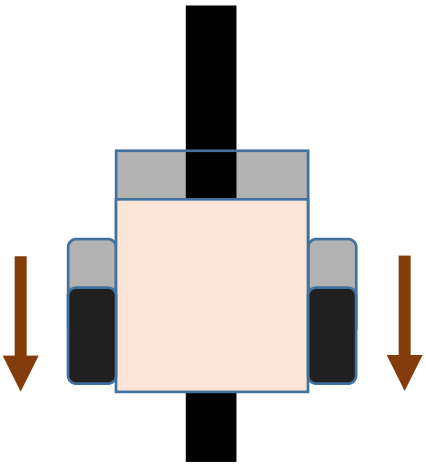
INTERCONNECTIONS OF ROBOT



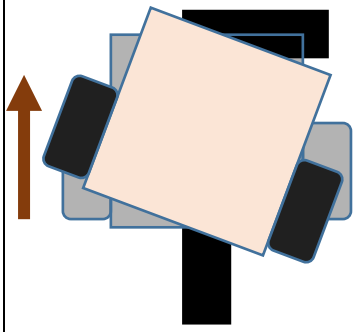
POSITION CONTROLLING



FORWARD

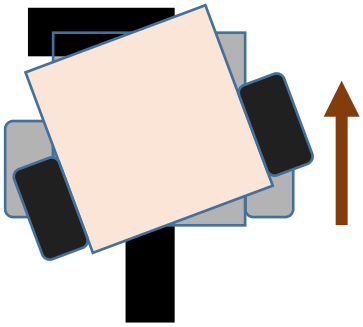


BACKWARD



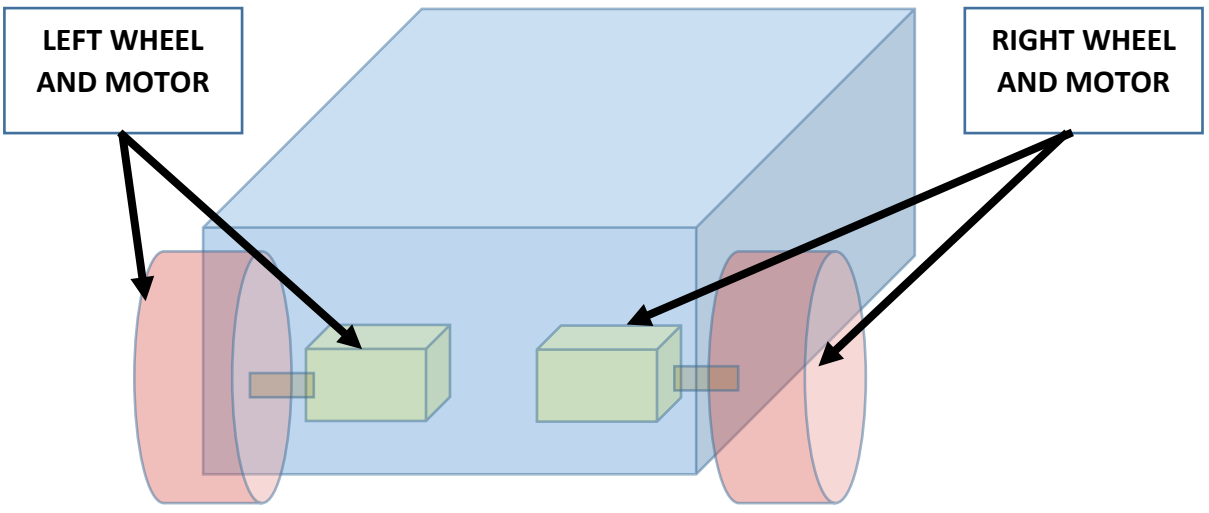
TURNING

RIGHT



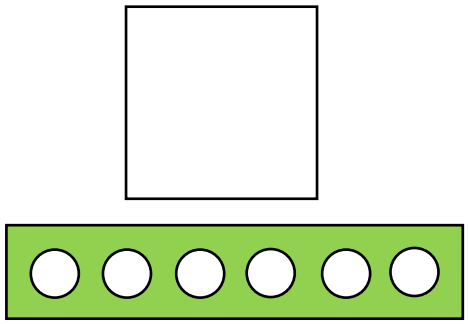
TURNING

LEFT

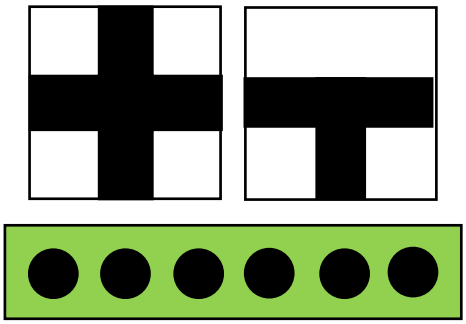


SENSOR PANEL ACTION

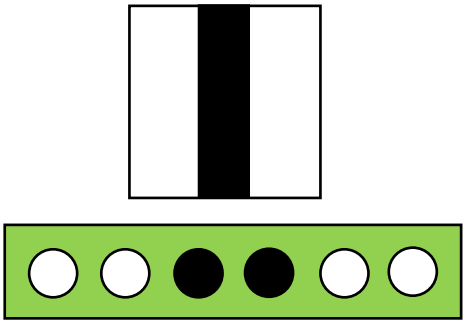
RIGHT SIDE  LEFT SIDE



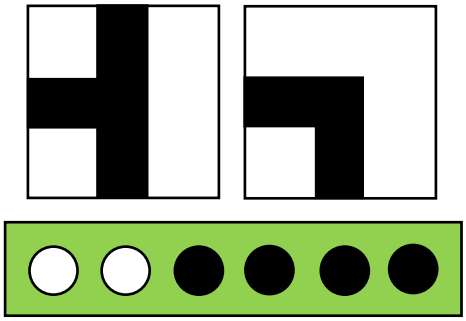
STATE 01



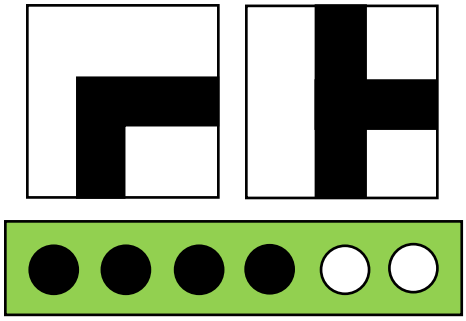
STATE 02, 03



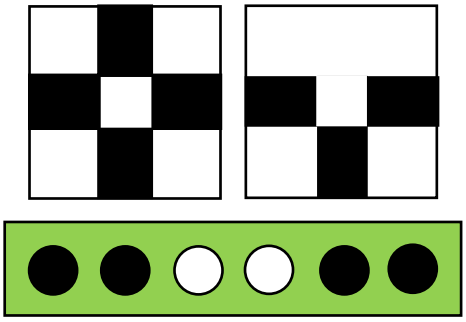
STATE 04



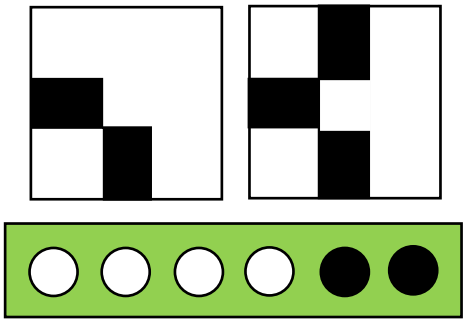
STATE 05, 06



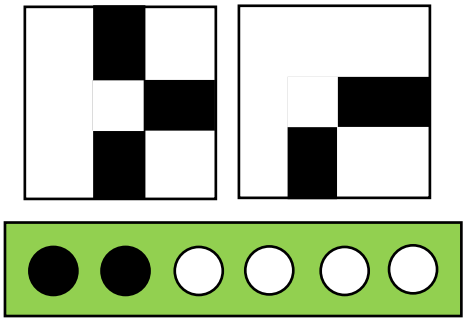
STATE 07, 08



STATE 09, 10



STATE 11, 12

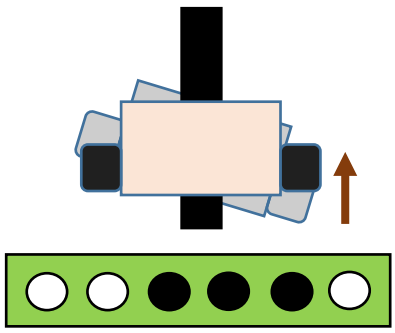


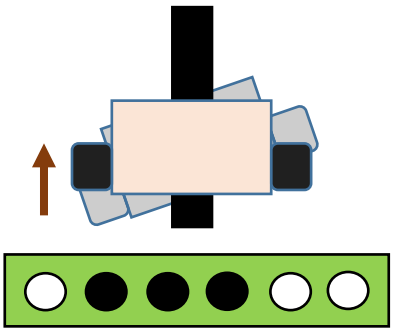
STATE 13, 14

SUMMERY OF THE FUNCTION

- **Line following**

  - When starting and moving on a line following is used,





  - Here, robot comes in to **state 04** by rotating its wheels to adjust the position.

- **Identifying junctions**

  - There 6 types of junctions in the grid. They are **state 02, 03, 05, 06, 07 and 08**.
  - **State 02, 03,** after these states sensor array shows **state 01,** the previous state is **state 02** and robot should turn right or left. If it is **state 04,** the previous state is **state 03** and robot should go forward.
  - **State 05, 06,** these are left hand junctions. After these states sensor array shows **state 01,** the previous state is **state 06** and robot should turn left. If it is **state 04,** the previous state is **state 05** and robot should go forward or turn left.
  - **State 07, 08,** these are right hand junctions. After these states sensor array shows **state 01,** the previous state is **state 07** and robot should turn right. If it is **state 04,** the previous state is **state 08** and robot should go forward or turn right.

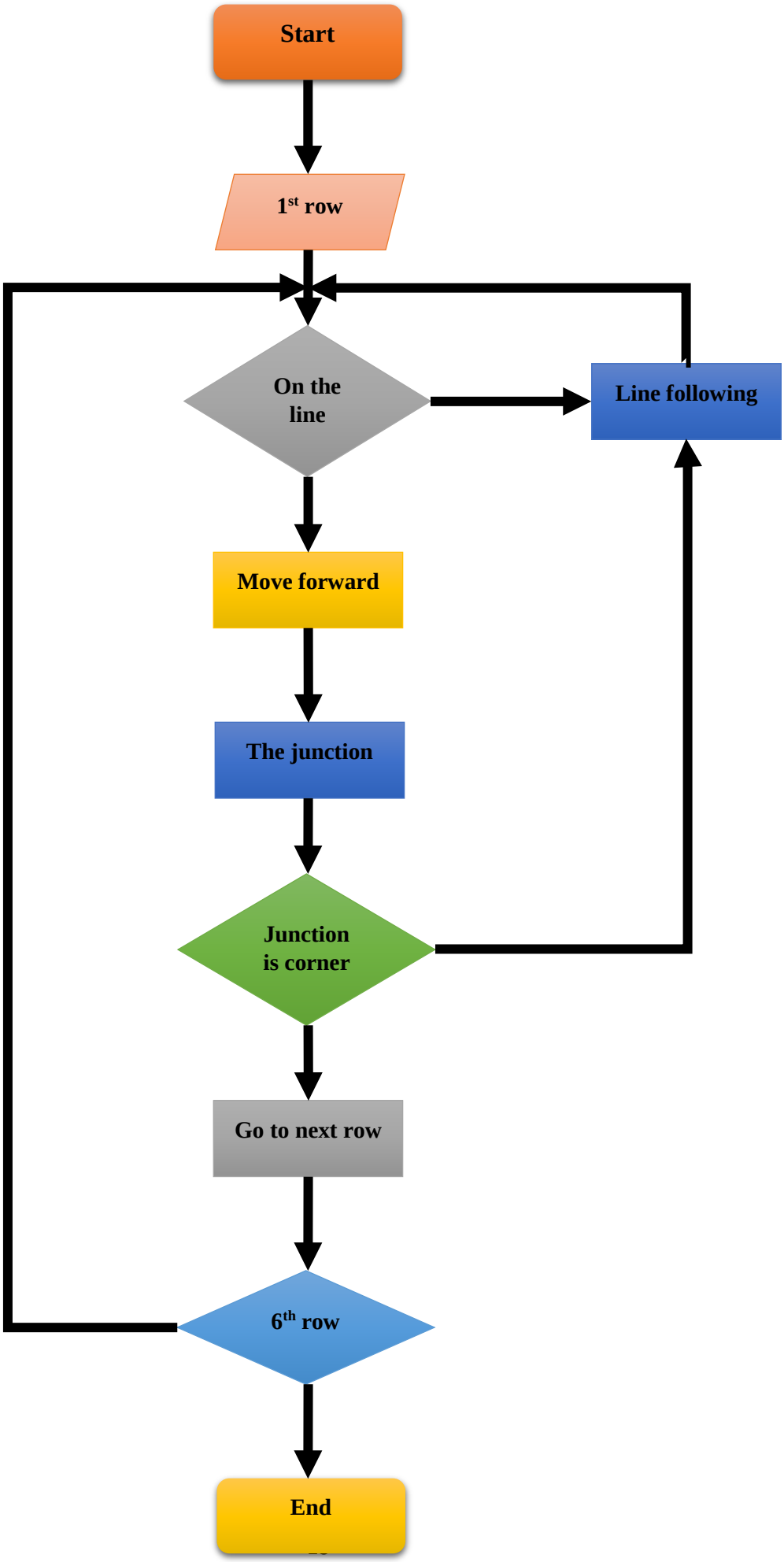
- **Identifying obstacles**

  - There 6 types of obstacles in the grid. They are **state 09, 10, 11, 12, 13 and 14**.
  - These obstacles are mapped while going through the grid.
  - These obstacles are avoided and complete the grid using shortest path in the next mode.

- **Mapping**

  - Mapping the grid is the function of identifying the location in a certain fixed data set and assigning a unique identification mark to locate them as they are referenced. In the above scenario the grid solving robot is expected to map all the point/nodes/junctions of the grid and input them into a matrix so that they can be read or processed when required. The grid solving robot identifies the location of the nodes and junctions based on the pattern it receives from the array on sensors. Using this information, it determines if it has reached a junction or a node. After this it further processes this information to check if it has passed over an obstacle. Obstacles give a unique data signature which can be used to vary the values of the matrix accordingly. For example, if all known junctions of the grid are assigned a value of **‘’1’’** in the matrix then the obstacles would be reassigned as **‘’0’’**. In this scenario the robot is expected to transverse a path between two fixed point using only the values **‘’1’’.** The essence of grid solving can be boiled down according to the following function.

ALGORITHM


Start
1st row
On the line
Line following
Move forward
The junction
Junction is corner
Go to next row
6th row
End

GRID SOLVING

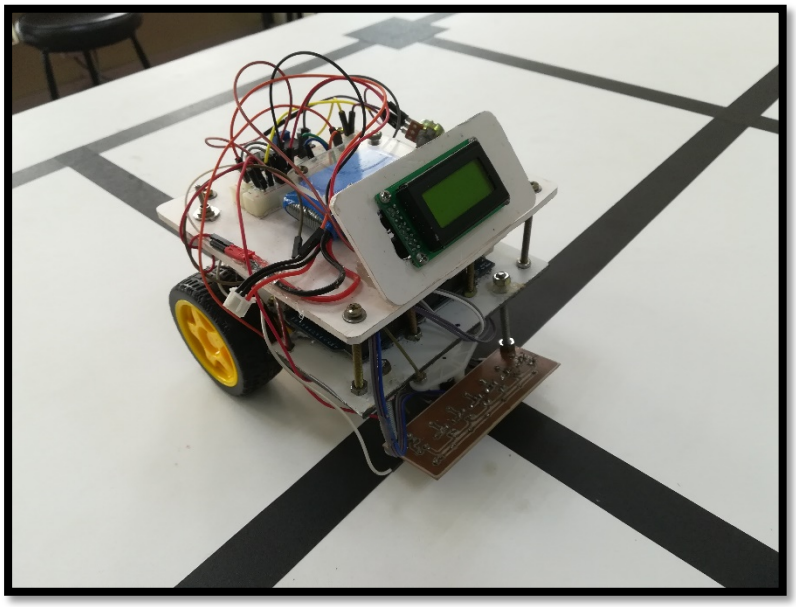
The first step of solving the grid is to obtain the required inputs correlating to the rows and columns of the two dimensional matrix. Once these values are obtained they are processed via an algorithm that selects the most suitable path to traverse. These set of operations are obtained during the first run where the robot travels across the whole grid and distinguishes the obstacles and the junctions. Once these have been identified the Arduino mega board gives the instructions required for the robot to start traversing the path. The necessary duration in which the wheels should turn is determined in real time by the input received from the IR sensor array. Using this data the robot decides in which direction it must transverse across the predefined grid and for how long it should travel before taking the next decision to turn, stop or move forward.

PROBLEMS AND DIFFICULTIES FACED

The main problems that were faced during this project can be divided into two sub sections, namely hardware and software. The main issues that were faced in the latter were problems related to the wheels not spinning at the same rate. This caused the robot to travel in a path which did not properly align according to the grid, it also meant that the micro controller had to constantly correct this small error. In effect a lot of vibration was noticed which prevented the proper orthogonal alignment of the sensor array with the line. The main issue that this caused was the inability to properly establish a definite set of patterns which could be referenced to make correct logical decisions. As a result programming the robot to transverse the grid was painstakingly difficult. To solve this issue the motors could be replaced with more reliable ones that had equal design and operation specifications. Software related issues the main problem that was faced during the programming of the robot was finding a proper code that was compatible with Arduino's software environment and architecture. Most of the codes that were researched into either ran in java with the inclusion of made different libraries which were unable to be translated to Arduino data/library environment. This meant that even simple functions like referencing the matrix were difficult because the proper libraries/data instructions did not exist in the arudino programming interface. To overcome this issue a suitable code was found which could be transferred properly to the Arduino board and successfully traverse the grid.

DISCUSSION AND CONCLUSIONS

The objective of this project was to create a fully equipped robot that has the ability of transversing a grid and successfully identifying the locations of obstacles which it cannot pass through. After it has accomplished the task of mapping the grid it is expected to find a path in which it can travel. It must have the ability of deciding which direction to turn and which directions to avoid. To tackle this problem an Arduino mega board has been used to solve the basic logic operations such as grid following and matrix processing. Grid following the action of following a grid is processed by the arduino mega board which interprets the input of an array of sensors. This array is positioned on the underside of the robot and acts as a continuous scanner which detects the location of the line. It accomplishes this by bouncing IR beams off the surface of the grid. The locations in which the line exists give a positive or "1" signature and the positions where it does give a "0" signature. Using this principle and with the aid of an array a virtual interpretation of line can be made. The Arduino board is employed to process this continuous stream of data and make the correct decision in order to follow the line. These instructions are then transferred to the motors which control the rotation of the wheels. Spinning the wheel at the same velocity moves the robot in the forward direction and spinning one wheel faster enables the robot to turn or rotate. This simple set of maneuvers is used to help the robot follow the line.



REFERENCES

- ijcjournal.org/index.php/InternationalJournalOfComputer/article/download/819/412 (Algorithm of line following robot)
- www.scribd.com/doc/35925045/Major-Project-Grid-Solving-Robot (final report)
- https://www.slideshare.net/udaywankar/grid-solving-robot (components of robot)