International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 2, Mar-Apr-2019, ISSN (Online): 2395-566X

Design and Implementation of a Robot for Maze-Solving using Wall Following Algorithm

K.Surya Teja Koyi Rajesh M. Gopi Krishna CH.Pavan Naga Sai
KoneruLakshmiah Educational Foundation
Guntur, A.P., India.
koyirajesh10@gmail.com

Abstract- **Maze solving robots in the past had great impact in robotics which is based on decision making algorithm, the main result of this project is to detect the objects and it should avoid them for colliding. if the object in on right and front then it should turn left, if the object is on left and front then it should turn right, if the object is on front and left and right then it should go back or stop there. Not only turning here we have decided to display the output in LCD if it turns right ,right is displayed in LCD and also same for left, back, front, stop. By doing this project we can solve the daily problems very quickly.**

Keywords **-Maze solving, Ultra sonic sensors, Arduino, Motor driver circuit, Artificial intelligence.**

I. INTRODUCTION

A maze is a network of paths, typically from an entrance to exit. The concept of Maze is on approximately thousand years old Which was invented in Egypt. From then, many mathematicians made various algorithm to solve the maze. Now a days, maze solving problem is an important field of robotics. It is based on one of the most important areas of robot, which is "Decision Making Algorithm". Cause, this robot will be placed in unknown place, and it requires to have a good decision making capability.

There are many types of maze solving robot using various type of algorithms. In this project, the system design of Maze solving robot consist obstacle avoidance ultrasonic sensors and then sensors will detect the wall. To solve the maze, this robot will apply wall following algorithms such as left or right hand rule. It will also follow the Flood fill algorithm for finding the shortest path. An autonomous robot is a category of robot that can perform tasks intelligently depending on themselves, without any human assistance . Maze Solving Robot is one of the most popular autonomous robots. It is a small self-reliant robot that can solve a maze from a known starting position to the center area of the maze in the shortest possible time. A maze solving robot make multiple runs in a maze, first it create a map of the maze layout and store it in its memory, then run through a shortest path .

However, both types of robots are designed to be autonomous, they basically perform different tasks. The maze solving robot designed in this tutorial is built on Arduino UNO and has the maze solving algorithm implemented within the Arduino Sketch. The hardwaredesign of the robot is quite similar that of any other typical line follower robot except that a line follower may have sensors only in the front side of the robot, the maze solving robot has sensors at left side, right side and front side of the robot. The electronic circuitry of the robot consists of the Arduino board, Ultrasonic sensor and L293D motor driver IC coupled with two geared DC motors. The robot is powered by a 12V battery and is programmed to instantly start finding an escape route once it is powered by the battery.

II. BLOCK DIAGRAM

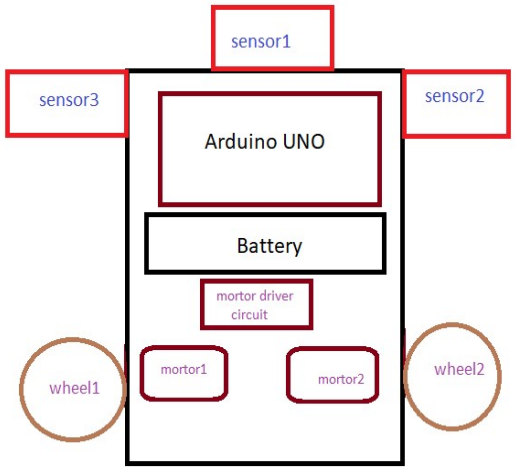

Fig.1 Block Diagram.

1. Ultra Sonic Sensor

Ultrasonic sensors emit short, high-frequency sound pulses at regular intervals. These propagate in the air at the velocity of sound. If they strike an object, then they are reflected back as echo signals to the sensor, which itself computes the distance to the target based on the time-span between emitting the signal and receiving the echo. The diagram of ultra sonic sensor is shown in below fig 2.

© 2019 IJSRET

International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 2, Mar-Apr-2019, ISSN (Online): 2395-566X

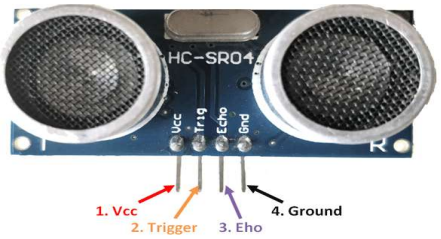

Fig.2 Ultra Sonic Sensor

Humans can detect sounds in a frequency range from about 20Hz to 20 KHz. However the frequency range normally employed in ultrasonic detection is 100 KHz to 50MHz. The velocity of ultrasound at a particular time and temperature is constant in a medium.
W = C/F (or) W = CT
Where W = Wave length
C = Velocity of sound in a medium
F = Frequency of wave
T=Time Period.

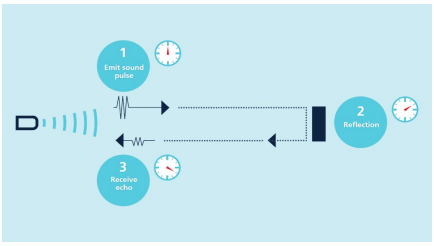

Fig.3 Object Detection

2. Ultra Sonic Object Detection

Ultrasonic sensors are used to detect the presence of targets and to measure the distance to targets in many robotized processing plants and process plants. sensors with an on or off digital output are available for detecting the presence of objects and sensors with an analog output which changes relatively to the sensor to target separation distance are commercially available and shown in the above fig 3.

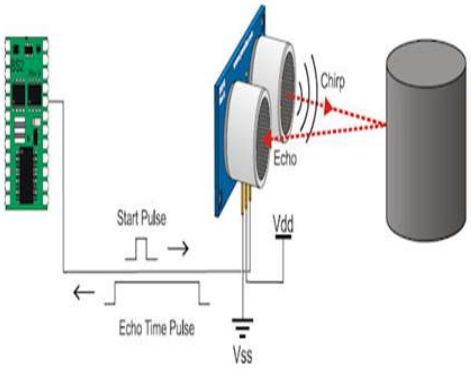

Fig. 4 Sensor Connection

3. The connection of the Ultrasonic Sensor to the Arduino

- 3 GND pins is connected to the Ground.
- 3 VCC pins is connected to the Positive (5v).
- Echo pin is connected to the Arduino. (choose any pin and match it to the code)
- TRIG pin is connected to the Arduino. (choose any pin and match it to the code).
- We are having total 3 ultra sonic sensors, so we should be careful while connecting each eco and trigger pins to Uno board.

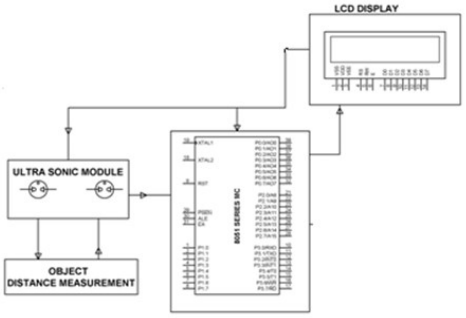

Fig. 5 Logisim Circuit.

4. Motor Driver Circuit

The L298 Driver is a high voltage, high current dual ful bridge driver designed to accept standard TTL logic levels and drive inductive loads such relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together the corresponding external terminal can be used for the connection of an external sensing resistor. The L298 integrates two power output stages (A,B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs.

The current that flows through the load comes out from the bridge at the sense output an external resistor (RSA, RSB.) allows to detect the intensity of this current. Each bridge is driven by means of four gates the input of which are In1, In2, EnA and In3, In4, EnB. The In input set the bridge state when The En input is high a low state of the En input inhibits the bridge. All the inputs are TTL compatible. A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the

© 2019 IJSRET

International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 2, Mar-Apr-2019, ISSN (Online): 2395-566X

GND pin of the IC. Each input must be connected to the source of the driving signals by means of a very short path. Turn-On and Turn-Off. Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable in-put must be driven to the Low state. L298N is an integrated circuit multi watt 15 package and capable of giving high voltage. It is a high current dual full-bridge driver which is designed as to accept standard TTL logic levels. It can drive inductive loads e.g relays, solenoids, motors (DC and stepping motor) etc.

5. Pin diagram of L298N motor driver

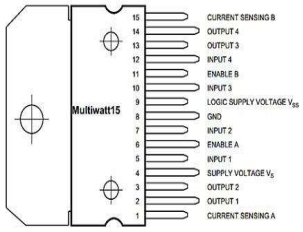

Fig.6 L298n Pin Description

6. L298N motor driver internal circuit

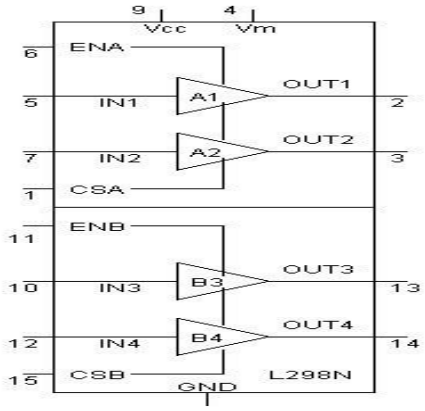

Fig.7 Internal Circuit

7. Charging and Discharging

During Charging, The Positive Active Material Is Oxidized,producing electrons, and the negative material is reduced, consuming electrons. These electrons constitute the current flow in the external circuit. The electrolyte may serve as a simple buffer for internal ion flow between the electrodes, as in lithium-ion and nickel-cadmium cells, or it may be an active participant in the electrochemical reaction, as in lead–acid cells. The energy used to charge rechargeable batteries usually comes from a battery charger using AC mains electricity, although some are equipped to use a vehicle's 12-volt DC power outlet.

The voltage of the source must be higher than that of the battery to force current to flow into it, but not too much higher or the battery may be damaged. Chargers take from a few minutes to several hours to charge a battery. Slow "dumb" chargers without voltage or temperature-sensing capabilities will charge at a low rate, typically taking 14 hours or more to reach a full charge. Rapid chargers can typically charge cells in two to five hours, depending on the model, with the fastest taking as little as fifteen minutes. Fast chargers must have multiple ways of detecting when a cell reaches full charge (change in terminal voltage, temperature, etc.) to stop charging before harmful overcharging or overheating occurs. The fastest chargers often incorporate cooling fans to keep the cells from overheating. Battery packs intended for rapid charging may include a temperature sensor that the charger uses to protect the pack; the sensor will have one or more additional electrical contacts.

Different battery chemistries require different charging schemes. For example, some battery types can be safely recharged from a constant voltage source. Other types need to be charged with a regulated current source that tapers as the battery reaches fully charged voltage. Charging a battery incorrectly can damage a battery; in extreme cases, batteries can overheat, catch fire, or explosively vent their contents.

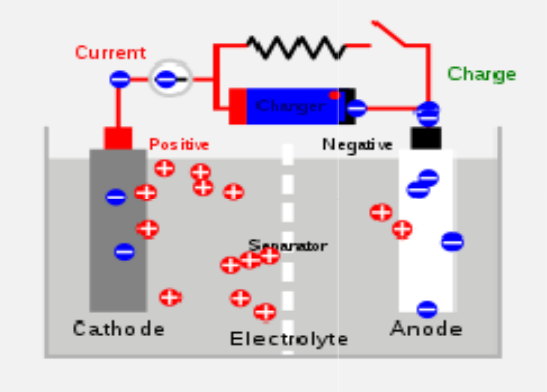

Fig.8 Charging and Discharging

II. METHODOLOGY

1. Wall Following Algorithm

The Most common algorithm for maze solving robot is Wall following algorithm. The robot will take its direction by following either left or right wall. This algorithm also called, left hand-right hand rules. Whenever the robot reaches a junction, it will sense for the opening walls and select it direction giving the priority to the selected wall. By taking the walls as guide, this strategy is capable to make the

© 2019 IJSRET

International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 2, Mar-Apr-2019, ISSN (Online): 2395-566X

robot reaches the finish point of the maze without actually solving it. But, this algorithm is not efficient method to solve a maze. Cause, the wall follower algorithm will fail to solve some maze construction, such as a maze with a closed loop region.

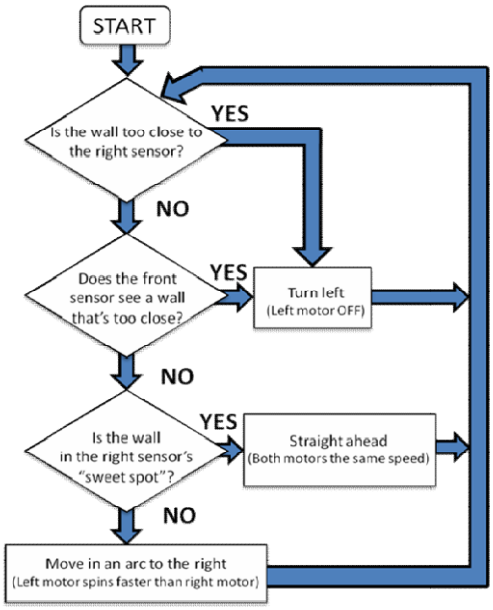

Fig.9 Flow chart for decision making.

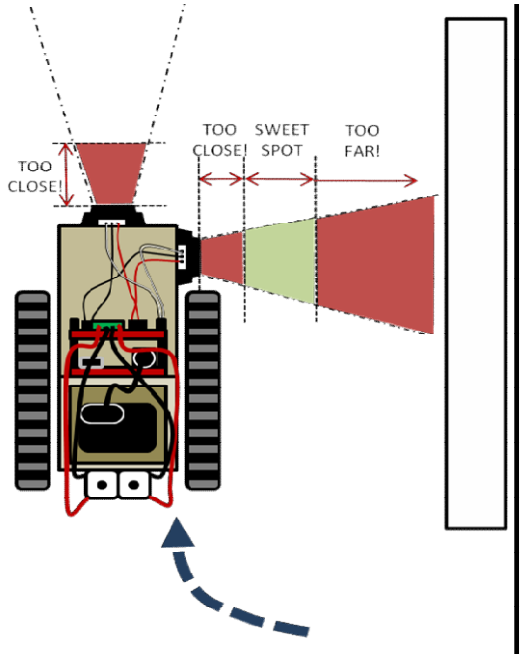

Fig.10 Wall Following Algorithm.

The robot will take its direction by following either left or right wall. This algorithm also called, left hand-right hand rules. Whenever the robot reaches a junction, it will sense for the opening walls and select it direction giving the priority to the selected wall. By taking the walls as guide, this strategy is capable to make the robot reaches the finish point of the maze without actually solving it. Here there is a Flow chart to achieve the required output, that is shown in the fig.9

III. EXPERIMENTAL RESULTS

By following the given algorithm we written code in embedded c and dumped into the robot. Here this robot works on the given algorithm, If the wall is too close to the right sensor then the left motor turned off and right motor rotates then the robot turns left. In the same way if the wall or obstacle is too close to the left sensor then the right motor stop rotating then the robot turns right. If the wall is too close to the front sensor then the left motor turns off and then robot turns left and at the same time if it founds wall on the left the right motor turns off and the robot turns right. If no obstacle is found by the sensors then the both motors works fast and robot moves fast. If the wall is in the right sensors sweet spot then both the motors works in same speed and robot goes straight ahead. Else the left motor spins faster than right motor and robot moves in an arc to the right. In this way the whole Mage solving robot working properly. Here is a table which shows motor performance in terms of RPM and no load current as a function input voltage.

Table2 Flow chart.

Voltage (V)	RPM (No Load)	Current (A)
4	16	0.013
5	21	0.014
6	28	0.014
7	36	0.016
8	42	0.0
9	47	0.019
10	52	0.020

Here is a table which shows Motor performance in terms of stall torque and stall current as a function of input voltage.

1. Final project

Here it is the final project photo after completion of total connections. This robot can detect the obstacles and move opposite according to the obstacles presence. This can

© 2019 IJSRET

solve the problems to find the way where people can notgo or any human being can not go.

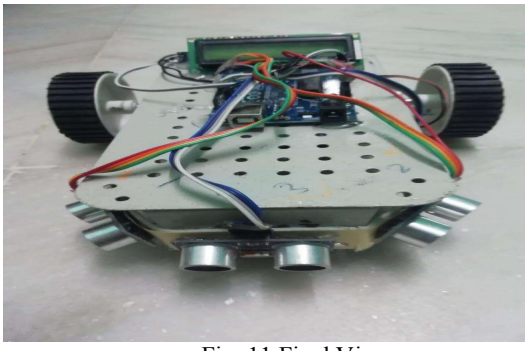

Fig. 11 Final View.

IV. CONCLUSION

As a conclusion, the maze solving algorithm have successfully been implemented in the robot and the objectives of the project have been achieved. The first algorithm was wall following algorithm. The basic method shows a good result for solving the maze. But, due to lack of self intelligence, it failed to solve the maze in the shortest way. And it could not solve to close loop maze. So, an efficient method has been used to find the shortest path that is flood fill algorithm method. After applying all methods, the robot was trained in a real maze. Several tests has been run to ensure the best performance of the robot. This project helps to improve various important information about robotics, knowledge about many decision making algorithms. It's also helped to learn about many electronics components such as motor driver, sensors, etc. This gained knowledge will have a significant impact on future work.From my opinion it is very interesting about mazes but we have to give correct commands and instructions.

REFERENCES

[1]. http://ewh.ieee.org/reg/1/sac/Main/RegionaIEvents/StudentConf/MicromouseRules.pdf
[2]. https://www.researchgate.net/publication/229000984_MacroMouse_A_Maze_Solving_Robot_Final_Report
[3]. Tondra De, Dfew Hall, "The Inception of Chedda: A detailed design and analysis of Micromouse", University of Nevada Las Vegas Senior Design, vol. 5, pp. 39-44, 2004.
[4]. HuoZhe Fu, "A Design and Realize for Micromouse's Maze-solving", Microcomputer Appfications, pp. 59-62, 2008
[5]. Kazerouni B.H. Moradi, "Variable Priorities in Maze-Solving Algorithms for Robot's Movement", Seville: Proceedings IEEE International Conference on Industrial Informatics, vol. 6, pp. 181-185, 2003.
[6]. http://www.eecs.ucf.edu/seniordesign/fa2014sp2015/g28/docs/SD2Report.pdf