ARDIUNO MAZE SOLVING

Mrs. Deshmukh Arati P [1], *Ms. Dabhade Priyanka A.* [2], *Ms. Lawande Pratiksha J.* [3], *Ms. Raut Vishakha V.*[4]

[1 ,] *Guide, Computer Engineering ,Marathwada Mitramandals Polytechnic , Maharashtra , India*
[2,3,4], *Student ,Computer Engineering ,Marathwada Mitramandals Polytechnic , Maharashtra , India*

ABSTRACT

Maze is the solving problem is a very difficult problem, but still now it is considered as an important field of module. This field is based on decision-implementation algorithms. The main working of this project is to make an Ardiuno based efficient maze solver model. Two simple mazes solving algorithms "Wall detection algorithm" are used to make this model. In this project Hardware development, software development and maze solving is to been done. For performance testing, the model will implement to solve the maze. The capability of finding the small path is also verified. maze solving model is a self-contained fully ardiuno maze solving model, it can capable of transport itself to the target of an unknown maze. The model must find its way to the target of the maze simply to return again and again in increasingly more success full attempts in order to certain the fastest run. The individual components of maze solving model system consist of the ardiuno motor control system, sensor array, and a mapping system or algorithm for the maze intelligently. The working of this research is to develop and implement a maze solving model and find the possible path from the starting point to the target by reducing the collision. This system can also reduce the errors by unbalancing the speed of the two motors. The main objective is to build fully maze solving model systems with complex environment. There are a number of techniques which have been used for solving the maze by model. In this research is used as path finding method to reach the target of the maze.

Keyword : - *Arduino , Ultrasonic Sensor, Motor*

1.Introduction

In this paper, the design ardiuno of maze solving model which has the ability to navigate automatically in an unknown area based on its own decision is presented. For the proposed design algorithm, a wall following technique based on LSRB and RSLB algorithm is used and front. The designed model obtains input from ultrasonic sensor,and wheel rotation encoders and then make decision for solving maze. It has the capability to solve the maze by taking the shortest distant path and it stores the details for the further reference also. The designed model has the ability to learn any arbitrary maze and find the possible shortest route for solving it. The best application of this designed model could be for navigational purposes Maze Solving model also called ardiuno solving is a model designed to get to the target of the maze, unaided. The model essentially comprises of a motor driver, steering and turning method to move the model. It has sensors to detect the wall and control logic to control the activity of the model and find the possible path. The model is powered by a battery and has to find its way from a predetermined starting point to the target of the maze unaided. The small the time period it takes, the better the decision for working algorithm. The model has to keep consider of where it is and detect once it reaches the finishing wall. The model will need to keep consider of where it is and detect when it has reached the goal. Ardiuno model have wide reaching applications, as reviewed in, from Bomb sniffing to finding well of maze. It is possible that a model can save lives and can be argued that not only saving a life of human but also saving the distance. This is why we have contributed to researches and advanced modeling studies. The modeling objective is to find the target of a well detection maze for run time the model will attempt to make its fastest run from the starting point to the last cells

1.1 Literature survey

maze solving model is virtually divided into twelve partitions. The model uses data such as its direction and its current location to change between exploring rules to save more time optimize the solving working process. A simulation was developed to verify the algorithm by using program. Test results show that partition central algorithm has higher average efficiency when compared to other algorithms path. In another study, the discretely assigned potential levels were discussed to the programming. Using these potential levels, the moter can effectively make ardiuno decisions to reach the goal. It demonstrates the method of assigning and controlling these artificial potentials to provide the most optimized root detect path.

2. Hardware Desing Of Maze Solving Model

In this research, the two-wheel differential drive maze solving model is composed of two 12V DC motors with each optical encoder. In the product consists of 1 model chassis with length 30cm and 20cm width, 2 wheels with a diameter of 66 mm, 1 piece ball caster and 12V DC motors which have been by the well as two pieces of the DC motor brack to pair on the chassis.In this maze solving model had 2 pieces rotary encoder. Rotary encoder attached to the DC motor to calculate the model roboting of the Dang and Song proposed An Efficient Algorithm for model Maze-Solving which was based on flood-fill algorithm but improved by reducing some steps not needed in certain cases. As there are channels in the maze where the model is forced to go only straight forward, when the robot is inside these channels, it does not need to perform all four steps of maze solving which is updating the wall, flooding the maze, determining which turn to be taken and moving to the next root. To reduce execution time, it only flood the maze when there is a turn by direction path. Partition-central Algorithm is another maze-solving algorithm where the standard unit wheel. The whole hardware system of this model can be seen in the block diagram.used three Ultrasonic sensors to detect maze wall at right, left and center position. Driver L293D module is controlled the direction of rotation and speed of a DC motor driver .Rotary encoder is used to calculate themodel rotation of the right, left center wheels. Push button was used to instruct the model to start. The system output would drive two DC motors driver that served as actuators to move the right and left wheels, so that the robot can move forward, turn to the right, turned to the left, and rotates revers direction is per working od model and path od well. Arduino controller serves to process the signal input from the three ultrasonic distance sensors, perform processing algorithms that is solved the possible path, and generates output signals to control a model.

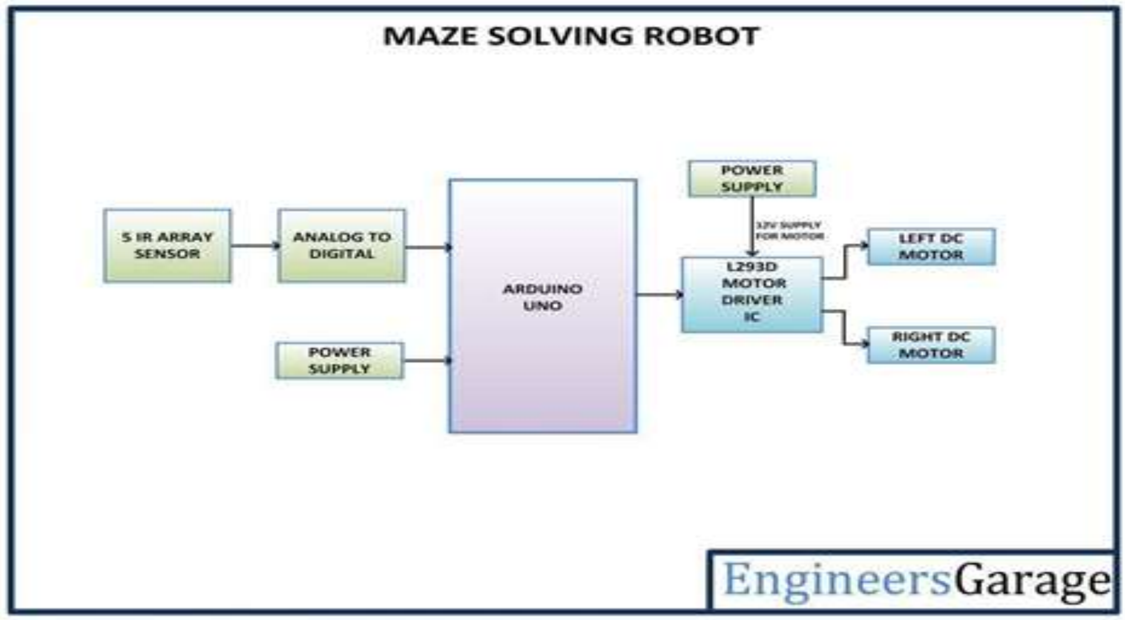

Fig-1: MODEL

Chart -1: **Maze Solving Design Approach**

In our design, it is important to focus to find ways and detect well in solving the maze to generate our own algorithms in combination with other types of algorithms that may be used for programming of our ardiuno controller. We have generated a method in solving a maze that combines 3 ultrasonic sensors in trying left, right and center possible paths through the maze in order to find the well destination. When does the model decide to turn? How does the model know what direction to turn? And how can we recognize the paths that are dead-end and omit them after the first round of root

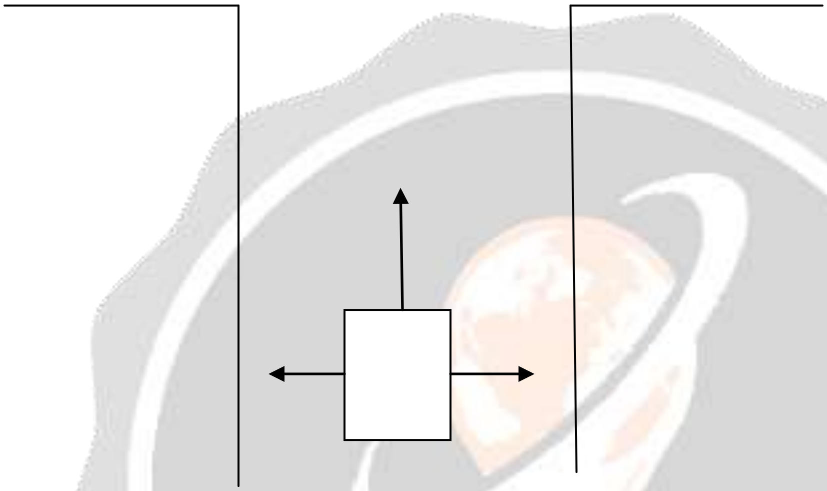

Fig-2: Root solving

Initially our model is moving centered between the three side walls. The left, right and center distance sensors detect these walls as the mouse the forward as shown in These sensors on the left and right also control the mouse to root through the center of the maze.At the straight forward move, the two DC motors drive same forward direction with the same. To move the model straight stably between walls, the distance from the right sensor and left sensor, center sensor need not to exceed more than 20 cm. If the left sensor exceed more than 10 cm from the left ultrasonic sensor, the model will move to the right to get stable move within the two wall. Same if the right sensor exceeds more than 10 cm from the right ultrasonic sensor, the robot will move to the left to get stable move within the two walls.The next step is that the sensor will navigate forward until it reaches the intersection or the position at the left sensor and right sensors center sensor cannot detect any walls as shown in When this happens, the left and right sensors center sensor will send a signal to the microcontroller telling the rdiuno controller to reduce the speed of the motors through the process of Pulse Width Model.As our model root forward, it will eventually reach a point where all the two side sensors (the left, right and center) does not detect any walls. At this point, the sensors send a signal to the ardiuno controller to reduce the speed of the robot to 1cm and begin the turning process at degree angle. Notice that the programming we are doing does not stop the model immediately, rather it will reduce the speed to 1 and turns 30-60 degrees toward the right or right according to the making of path finding algorithm. If the ardiuno controller decided to turn the left for the robot based on the algorithm, the motor at the left is moved reverse movement and center is moved reverse movement the motor at the right moves forward movement. That movement makes the model to left, center direction turns as discuss

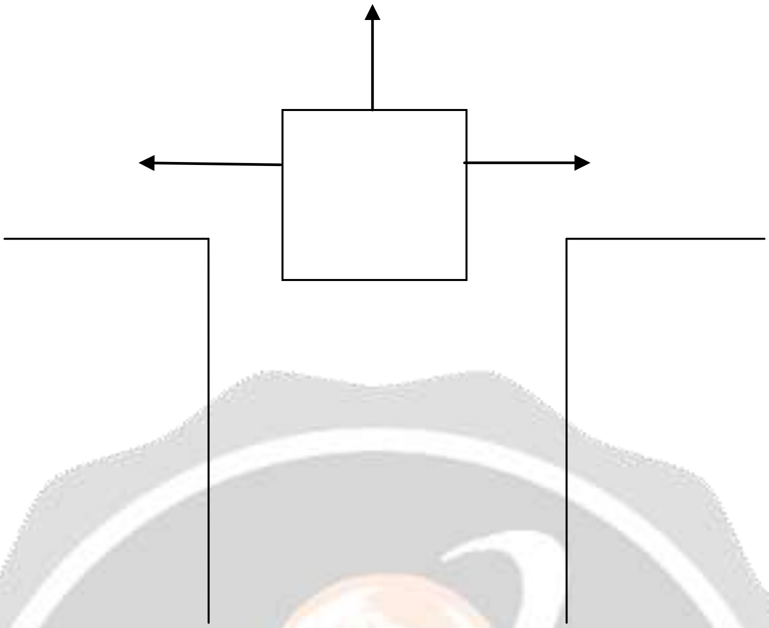

Fig-3: Left,Right Root searching

Similarly, the ardiuno controller is decided to turn the right for the robot based on the algorithm, the left motor drive forward direction movement and reverse direction movement is driven at the right motor of the model. Therefore, the maze solving robot turns to right root direction. The model may root through a route that is dead-end. In this case, all three side sensors, the left and right sensors as well as the center sensor detects wall When this scenario occurs, there will be a model will stop and brakes before hitting the wall in the range of sensor detection we have specified by design . Then, the model stops, it will make a 90 degree turn to turn around. The model is capable of making the 90 degree turns easily. At that condition, the two motor moving forward and reverse root direction.

2.1 Turning Characteristics of wheel

By changing the speeds of the two drive wheels independently each of the required can be performed. When both motors are model at the same speed the model will theoretically travel in a straight well root. The relationship between the two distances travelled by each of the wheels during turning as per root path. Using this method a simple mathematical formula can be employed to calculate the ratio of the outside wheel the inside wheel velocity using the ratio of the distance travelled by each of the wheels properly work on well path

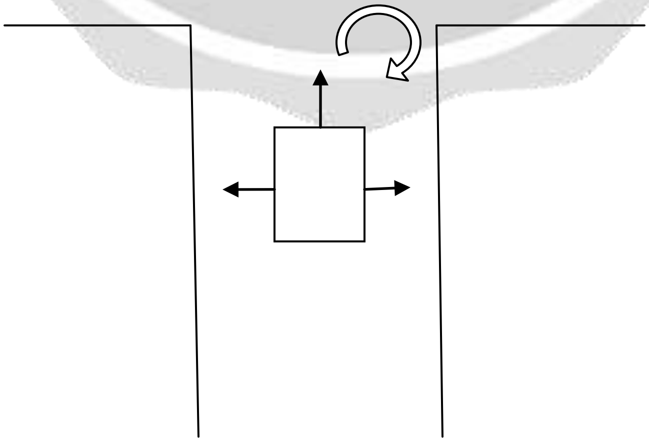

Fig-4: Turn the model bu using root

As the each wheel is theoretically identical, this ratio corresponds direly to the velocity of rotation for each of the wheels and motors driver. The greater the ratio, the sharper the turning curve performed by the ardiuno controller. Using this ratio, precise turning control of the robot can be achieved by altering the velocities of the motors driving dc the wheels.

2.2 The maze and the model

The maze designed for the model to solve is of the size of The maze was designed so that it will have two paths in order by programming for it to be solved. One of the paths is longer than the other path. The model must decide which one of the paths is suitable and solve the maze through that pathChoosing an algorithm for the maze model is critical in solving the maze. In this research, flood-fill algorithm was chosen well to solve the maze due to its balance in efficiency and complexity of root. There are four main steps in the algorithm: Mapping, Flooding Updating and Turning detecting which are described in the

Fig-5: Maze solving

3. Mapping the maze

For the model to be able to solve the maze, it has to know how big the maze is and virtually divides them into certain number of cells that can be used later in calculating the possible to the root of path destination. Between two cells there can be a walls. Thus, in a row of cells, there are 8 walls in between them. In total, in a row, there are eleven units of cells or walls direction . This information is stored in an array. The white units are the cells which the model can be placed inside. The orange units are the root for potential walls. The black units indicate wall intersections which are ignored by the algorithm of program. The external borders of the maze are also ignored as they are fixed boundaries of the maze as per root. Both cells (white) and walls (white) are set to zero in as their initial conditions Updating the wall data

3.1 Hardware requirement
1. Ardiuno controller
2. Moter driver(l293d ic)
3. Ultrasonic sensor(hc- sr04)
4. Wirse and cable
5. Wheels
6. Dc moter driver

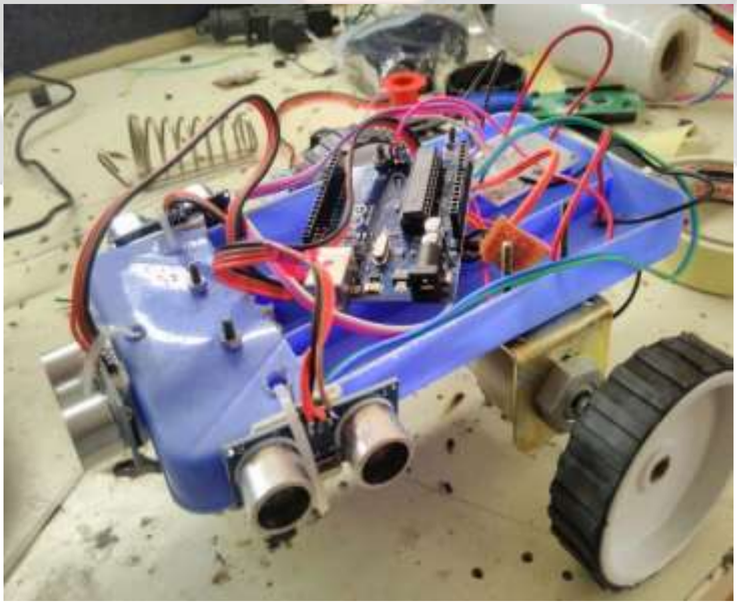

Fig-6: model of maze solver model

Fig-7: working of model

4. CONCLUSIONS

As a conclusion, the two mazes solving algorithm have successfully been implemented in the model and the objectives of the project have been achieved. The first algorithm was following algorithm. The basic method shows a good result for solving the arduino maze. But, due to lack of selfintelligence, it failed to solve the maze in the shortest way by using model. And it could not solve to close root maze. So, an efficient method has been used to find the shortest path or root that is flood fill algorithm method. After applying all methods, the model was trained in a real maze. Several tests has been run to ensure the best performance of the model. This project helps to improve various important information about robotics, knowledge, make model about many decision making algorithms. It’s also helped to learn about many electronics components such as motor driver, sensors, wires etc. This gained knowledge will have a significant impact on future work.

5. ACKNOWLEDGEMENT

Firstly, I would like to express my gratitude to God for providing me health, patience and knowledge to successfully complete this project. I submit my highest appreciation to my supervisor Dr. David Southall for his beneficial advices, support and guidance throughout this project. A special thank for all academic staff whose makes me reach this level of education. Thanks to all technicians staffs for providing me the required equipments and component. I would also like to thanks the Ministry of Education, Kingdom of Bahrain for there sponsorship and for providing me the opportunity to completing my studies in a good university. Finally, thanks to my parents, family and friends for their help, encouragement and guidance, who without I would not be where I am today

6. REFERENCES

[1] Sandeep Yadav, Kamal Kumar Verma, Swetamadhab Mahanta, “The Maze Problem Solved by Micro mouse”,International Journalof Engineering and Advanced Technology (IJEAT),ISSN: 2249 – 8958, Volume-1, Issue-4, April 2012

[2] Ibrahim Elshamarka, Abu Bakar Sayuti Saman, “Design and Implementation of a Robot for Maze-Solving using Flood-Fill Algorithm”, International Journal of Computer Applications (0975 – 8887),Volume 56– No.5, October 2012

[3] Wyard-Scott, L. and Q.H.M. Meng, “A potential maze solving algorithm for a micromouse robot”, Communications, Computers, and Signal Processing, IEEE Pacific Rim Conference on 1995 [4] George Law, “Quantitative Comparison of Flood Fill and Modified Flood Fill Algorithms”,International Journal of Computer Theory and Engineering, Vol. 5, No. 3, June 2013

[5] Babak Hosseini Kazerouni, Mona Behnam Moradi and Pooya Hosseini Kazerouni, “Variable Priorities in Maze-Sloving Algorithms for Robot’s Movement”, 2003.

[6] Sandeep Yadav, Kamal Kumar Verma, Swetamadhab Mahanta, “A New Method for a Micromouse to Find its Way through a Maze Unaided”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-4, April 2012

[7] Chang Yuen Chung, “Micromouse maze solving robot” (Universiti Teknologi Malaysia, May 2009)

IJARIIE