Autonomous Maze Solving Robot

Gan Zhen Ye , Dae-Ki Kang[1)]
Div. of Computer and Information Engineering, Dongseo University
Email : zy_gan@msn.com, dkkang@dongseo.ac.kr

ABSTRACT

Autonomous robots are intelligent machines that are capable of performing task in the world themselves with little or no human intervention. One of the main reason autonomous robots gained popularity in human's world is their ability to perform task with high degree of precision, accuracy and also consistency. One of the most studied fields in autonomous robot is the ability of decision making in robots.

To tackle the ability of robots to make decision, this paper proposed an Autonomous Maze Solving Robot that is able to solve a maze using the optimum solution. The maze and the design of the robot are in compliance with IEEE Micromouse competition rules and regulation. Micromouse is an autonomous maze solving robot that shall be able to explore a maze on its own from a predefined starting location and find the optimum path to reach the predefined goal in the maze without human's intervention.

Keywords

Autonomous robots, Maze Solving, Micromouse

I. Introduction

Autonomous robots are intelligent machines that are capable of performing task in the world themselves with little or no human intervention. One of the main reason autonomous robots gained popularity in human's world is their ability to perform task with high degree of precision, accuracy and also consistency. One of the most studied fields in autonomous robot is the ability of decision making in robots.

To tackle the ability of robots to make decision, this paper proposed an Autonomous Maze Solving Robot that is able to solve a maze using the optimum solution. The maze and the design of the robot are in compliance with IEEE Micromouse competition rules and regulation. Micromouse is an autonomous maze solving robot that shall be able to explore a maze on its own from a predefined starting location and find the optimum path to reach the predefined goal in the maze without human's intervention.

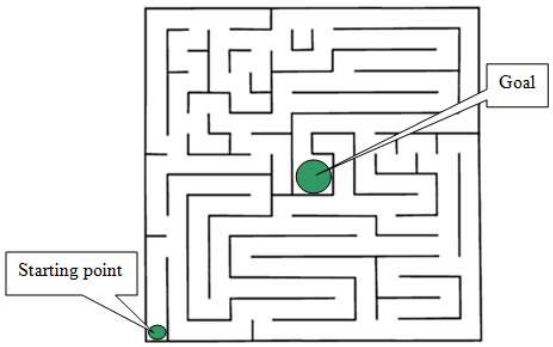

Fig. 1 Example of a Micromouse Maze

II. Hardware design

The maximum dimension allowed for a Micromouse is 25cm (0.25m) by 25cm (0.25m). However, the space available in a square unit of the maze is only 16.8 (0.168) as shown in Figure 2. Hence we designed the Micromouse to be as small as possible for it to be able to maneuver easily in the maze and also be able to perform a 180 degree rotation in one square unit of the maze without having any contact with the walls of the maze. This enables the mouse to escape from a dead end easily.

1) Corresponding author

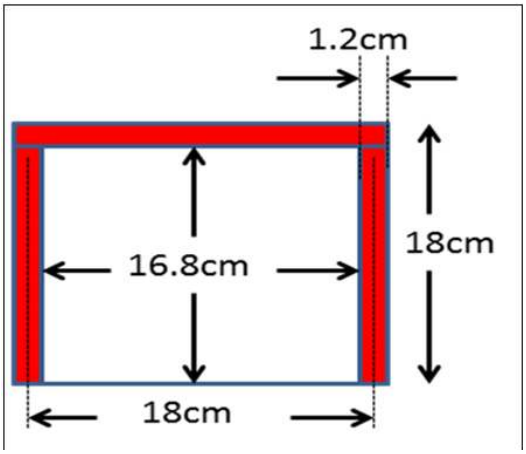

Fig. 2 Dimension of one square unit of the maze

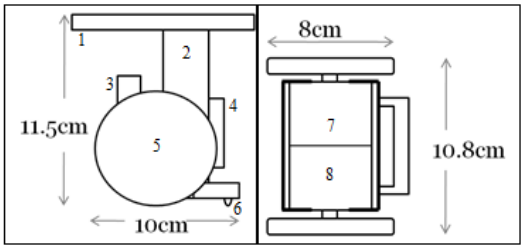

Fig. 3 Side view and top view (without controller board) of Micromouse

Figure 3 shown above is the design of the Micromouse, with each part labeled and listed down as shown below:

1. Controller board
2. Battery + aluminum casing
3. Aluminum casing for stepper motor
4. Sensor board
5. Plastic wheel
6. Castor wheel
7. Stepper motor
8. Stepper motor

The controller board consists of a PIC18F452 as the controller, a voltage regulator circuit, and a stepper motor driver circuit which uses 7404 (NAND gate) and 7486 (XOR gate). There are 4 IR sensors on the sensor board, with 2 sensors pointing towards the front direction and the other 2 sensor each pointing to the left and right direction. The purpose of having 2 sensors pointing forward is such that the mouse can know if it is moving in a straight path by comparing feedback values from both sensors.

III. Firmware design

The control architecture used is deliberative control which uses a Sense-Plan-Act paradigm. The algorithm used in solving the maze is a combination of conventional and modified flood-fill algorithm. The overall firmware flowchart is as shown in Figure 4 below.

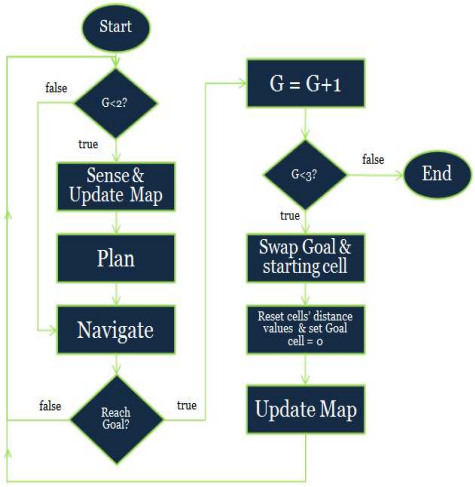

Fig. 4 Firmware flowchart

Referring to Figure 4, modified flood-fill algorithm is used in the planning stage to calculate the cells' distance from the goal. Conventional flood-fill algorithm is applied only right after the mouse reaches the goal. The difference between these 2 methods are conventional flood-fill algorithm will check and update each and every cell's distance value from the goal while the modified flood-fill will only update the cell's distance value if its neighboring cells' values are changed. An example of a maze with each cell's distance values from the goal can be seen in Figure 5.

IV. Conclusion

This autonomous maze solving robot is able to explore the maze and find the optimum solution path to solve the Micromouse maze without human intervention.

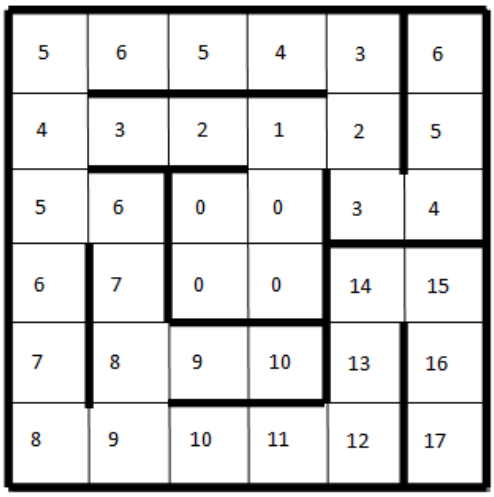

Fig. 5 Example of a flooded maze

Acknowledgment

This research has been conducted in the project, "A study on HTML & Javascript engine analysis and their extension plan", which has been commissioned by Electronics and Telecommunications Research Institute's project "Development of collaboration service technologies among smart screens using dynamic relocation of Web fusion contents" in 2011.

Reference

[1] Swati Mishra, Pankaj Bande, "Maze Solving Algorithms for Micro Mouse," in Proc. of 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems, Nov. 2008, pp. 86-93.

[2] Jianping Cai, Xuting Wan, Meimei Huo, and Jianzhong Wu, "An Algorithm of Micromouse Maze Solving," in 2010 10th IEEE International Conference on Computer and Information Technology (CIT 2010), June 2010, pp. 1995-2000.

[3] Wyard-Scott and H.M. Meng, "A Potential Maze Solving Algorithm for a Micromouse Robot," in Proc. IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing, May 1995, pp. 614-618.

[4] Tak Auyeung. (2005, June 10). Micromouse Handbook. [Online]. Available: http://www.mobots.com