Sensors & Transducers, Vol. 168, Issue 4, April 2014, pp. 120-126

Sensors & Transducers

http://www.sensorsportal.com

Turnings and Its Calibration of Micromouse Based on ARM9 and FPGA

[1, 3] Zhang Haoming, [2] PEH Lian Soon, [3] Wang Yinghai

[1] Department of Electrical and Information Engineering, Tongling University, Anhui Province, 214000, China

[2] Shining Sunshines Global Pte Ltd, Singapore

[3] School of Electrical and Information Engineering, SIPIVT, Jiangsu Province, 215123, China

[1] Tel.: 13013801792

[1] E-mail: zhm20060616@163.com

Received: 19 January 2014 /Accepted: 7 March 2014 /Published: 30 April 2014

Abstract: Micromouse is an intelligent robot that is designed to search a shortest path to the destination in an unknown maze. In order to meet micromouse special requirements: rapidity and stability, traditional single core micromouse control system is improved, and dual core control system based on ARM9 and FPGA is proposed, stepper motor is substituted by DC motor. In order to improve micromouse competition results, reduce run time of micromouse in complex maze, fast turning principle of micromouse is analyzed in detail, using different parameters to control its acceleration, speed and position, and other parameters to calibrate its attitude, which can ensure the rapidity, accuracy and good stability of micromouse in high speed turning. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Micromouse, Maze, Turning, Calibration.

1. Introduction

Micromouse championship is an international robotics competition, which micromouse solves a 16x16 maze. The maze is made up of 256 squares arranged in a 16 by 16 grid of cells, each 180 mm square with walls 50 mm high and 12 mm thick, the length of the wall is 168 mm. The micromouse is completely autonomous robots that must find their way from a predetermined starting position to the central area of the maze unaided. Having reached the goal, the micromouse will typically perform additional searches of the maze until it has found an optimal route from the start to the centre. Once the optimal route has been found, the micromouse will run that route in the shortest possible time [1-8].

The first world micromouse competition was held in Tsubuka, in Japan, in August 1985, and was open to contestants from Europe and the USA. The micromouse was now becoming quite sophisticated, using infra-red or ultrasonic sensors, stepper or DC servo-motors. Now the events are held worldwide, and are most popular in the UK, US, Japan, Singapore, India and South Korea [9-10].

The basic function of a micromouse is to travel from the start square to the destination square. This is called a run. The time it takes is called the run time. Travelling from the destination square back to the start square is not considered a run. The total time from the first activation of the micromouse until the start of each run is also measured. This is called the maze time. If a mouse requires manual assistance at any time during the contest it is considered

Article number P_1966

touched. By using these three parameters the scoring of the contest is designed to reward speed, efficiency of maze solving, and self-reliance of the micromouse [11-13].

Different countries use different score recording standards, the most representative standard is:

The scoring of a micromouse shall be obtained by computing a handicapped time for each run as follows:

Score Time = Run Time + search Penalty + touch penalty.

Search penalty = 1/30 of the maze or search time, in seconds, associated with that run.

Touch penalty = 3 seconds plus 1/10 of the run time, in seconds, if the micromouse has been touched at any time prior to the run.

Can be seen from the international standard, the main time of micromouse takes is run time and penalty. Because complexity of the technology, there are few domestic research institutions or teams engage in researching and designing it, micromouse development level is relatively out-of-date, designed micromouse principle as shown in Fig. 1.

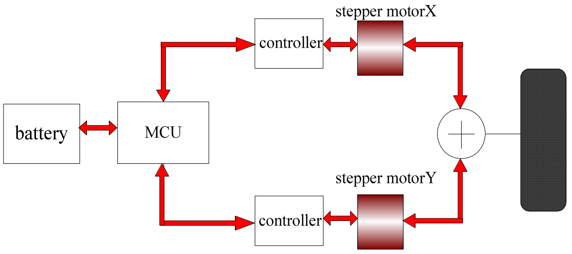


Fig. 1. Diagram of traditional micromouse principle.

After a long time operation found that the system has many problems: the micromouse is controlled by stepper motor, which often encounters the problem of missing drive signal, leading to the micromouse memory an error maze position, sometimes the micromouse can't find the dashing destination; no auxiliary angular velocity sensor help it in turning, often leads to a too small or too much turning angle, then only rely on compensation sensor to correct its moving, when meets continuous turning in complex maze, especially in high speed exploration and dashing, the micromouse may crash with the wall, resulting in a failure searching and dashing.

Therefore, the existing micromouse controller based on single chip needs to improve.

In order to overcome the problem of a single chip micromouse can't meet the requirements of stability and rapidity in fast turning, the paper abandon domestic micromouse single chip operating mode, referee to foreign modern design ideas, a new control mode based on ARM9 (S3C2440A) and FPGA was researched and developed [14-15]. The controller takes FPGA as the data handling core, to realize real-time two DC servo motor digital signal control in fast turning, ARM9 (S3C2440A) is free from the complex work, only realize logic control and a few calculations.

Principle of dual core micromouse as shown in Fig. 2.

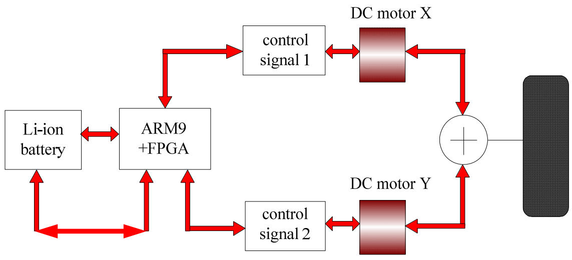


Fig. 2. Diagram of improved micromouse principle.

2. Turning Types of Micromouse

There are two main types of turns that the robot will have to perform when it is exploring the maze which are smooth turnings and stationary turnings.

The smooth turnings are left and right 90° turns (as shown in Fig. 3).

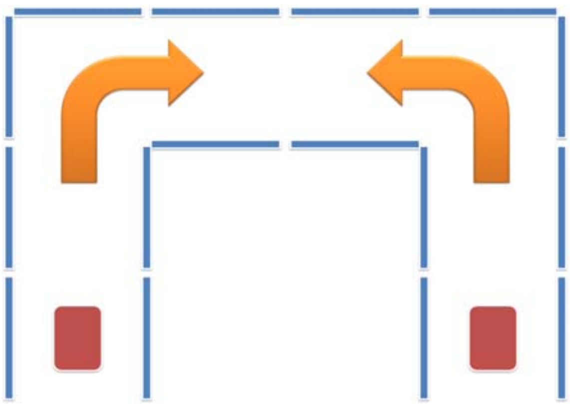
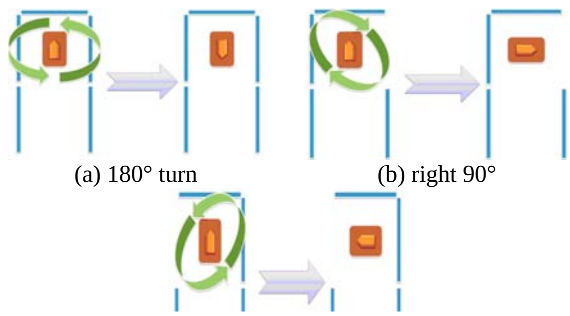


Fig. 3. Left and right 90° smooth turns.

The stationary turnings are 180°, right 90°, and left 90° turns (as shown in Fig. 4).



(a) 180° turn (b) right 90°

(c) left 90°

Fig. 4. Stationary turns.

3. Smooth Turnings of Micromouse

If the robot is moving and the next motion command is either a left or right turn, then it can perform a smooth turning. The smooth tuning is achieved by changing the outer wheel's velocity to a larger while and decreasing the inner wheel's velocity and that is how the robot turns, its software flowchart as shown in Fig. 5.

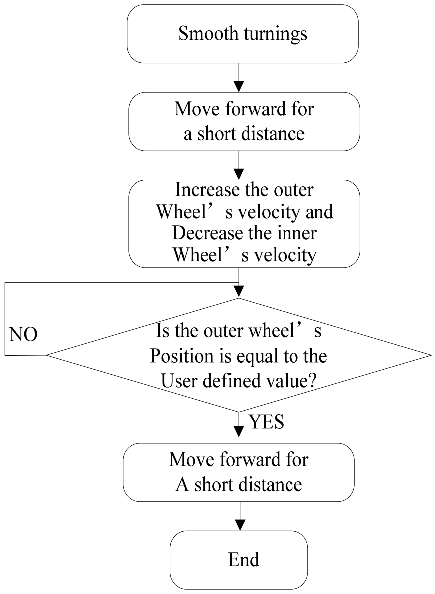


Fig. 5. The flowchart of smooth turnings.

The flowchart above shows that the robot will have to move forward for a short distance in order to turn in the centre of cell before it starts to change the velocities for the two motors. And the routine will keep checking the outer wheel's position and it will stop turning by changing back to the same velocity for the two motors when the outer wheel's position is same with the user defined value. After that, the robot will have to move forward for a short distance again in order to reach the next cell.

4. Stationary Turnings of Micromouse

If the robot comes to a dead end, then it has to perform a stationary 180° turn in order to continue its exploration. When the way the solver plans to go is blocked, the motion command execution routine will request the solver to generate a new command list and if the robot is going to crash with an obstacle while the solving is running, then it will perform an emergency stop and wait for the new motion command list. Under these two conditions, if the next motion command is a turning command, then the robot will perform a stationary 90° turn instead of a smooth turn. When performing a stationary turn, the two motors will run at the same speed but with the different direction, as shown in Fig. 6.

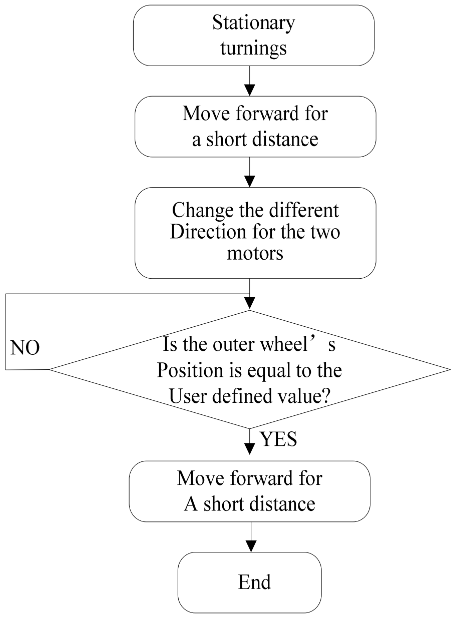


Fig. 6. The flowchart of stationary turnings.

The flowchart above shows that the robot will have to move forward for a short distance in order to turn in the centre of cell before it starts to change the direction for the two motors. And the routine will keep checking the outer wheel's position and it will stop turning by changing back to the same direction for the two motors when the outer wheel's position is same with the user defined value. After that, the robot will have to move forward for a short distance again in order to reach the next cell.

5. Calibration of Micromouse Normal Turn

This section for basic turning of micromouse, which includes left turn and right turn. In order to ensure accurate turning of micromouse, the paper use the following parameters to control the turning: distance before a turn, distance after a turn, turn angle, velocity, front correction, etc. Normal turn calibration software designed by VB as shown in Fig. 7.

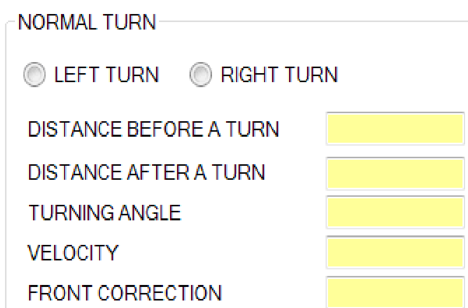


Fig. 7. Normal turn calibration software by VB.

5.1. Distance Before a Turn

After the micromouse sensed a changed in the sensor value (high value-low value), it will run for a certain distance before performing the turn, as shown in Fig. 8.

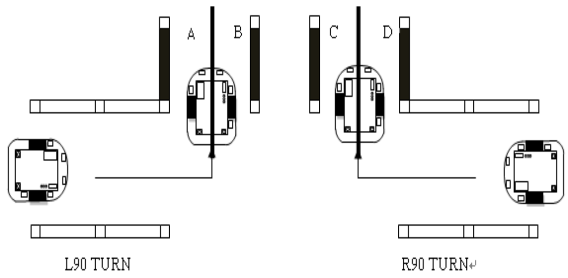


Fig. 8. Desired distance before a turn.

When micromouse make a left 90° turn, the micromouse should lie in between the wall highlighted in black as desired. If micromouse doesn’t lie in the desired position after the turn, can change the value according the following rules:

1) Increase the value if the micromouse is nearer to point A.

2) Decrease the value if the micromouse is nearer to point B.

When micromouse make a right 90° turn, the micromouse should lie in between the wall highlighted in black as desired. If micromouse doesn’t lie in the desired position after the turn, can change the value according the following rules:

1) Increase the value if the micromouse is nearer to point D.

2) Decrease the value if the micromouse is nearer to point C.

5.2. Distance After a Turn

After the micromouse done a turning, it will run for a certain distance before run straight or do another turning again, as shown in Fig. 9.

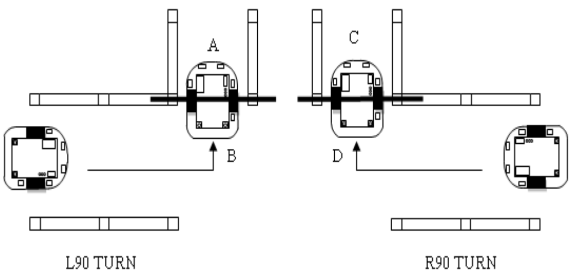


Fig. 9. Desired distance after a turn.

When micromouse make left 90° turn, after the turn, the wheels of the micromouse should align with the black line (beam-to-beam). If micromouse doesn’t align with the desired position after the turn, can change the value according the following rules:

1) Decrease the value if the micromouse stops nearer to point A (After the line).

2) Increase the value if the micromouse stops nearer to point B (Before the line).

When micromouse make a right 90° turn, after the turn, the wheels of the micromouse should align with the black line (beam-to-beam). If micromouse doesn’t align with the desired position after the turn, can change the value according the following rules:

1) Decrease the value if the micromouse stops nearer to point C (After the line).

2) Increase the value if the micromouse stops nearer to point D (Before the line).

5.3. Turning Angle

Turning angle is the angle of the turn performed by the micromouse. Perfect values of turning angle (Both L90 and R90 turn) can make the microumouse be in the proper position, as shown in Fig. 10.

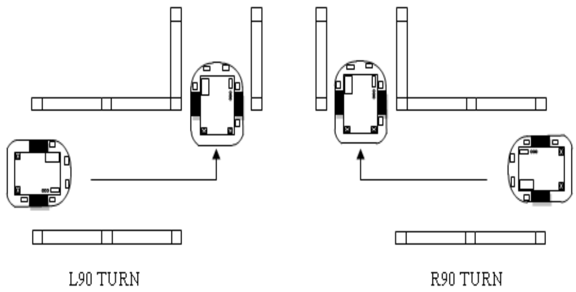


Fig. 10. Desired turning angle.

If the angle of the turn performed by the micromouse more than the ideal position value, the micromouse will make an over turning, as shown in Fig. 11. In this case, decrease the angle of the turn performed by the micromouse in L90 turn or R90 turn, which can correct its position.

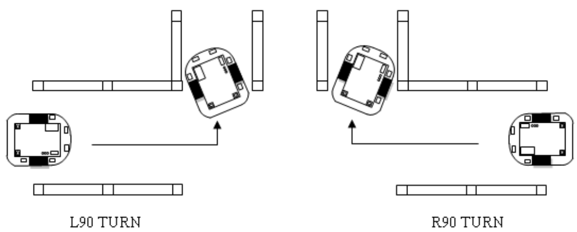


Fig. 11. Over turning angle.

If the angle of the turn performed by the micromouse less than the ideal position value, the micromouse will make a under turning, as shown in Fig. 12. In this case, increase the angle of the turn performed by the micromouse L90 turn or R90 turn, which can correct its position.

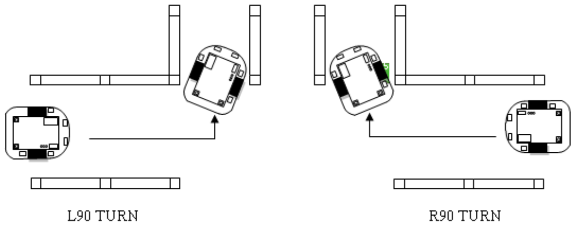


Fig. 12. Under turning angle.

5.4 Turning Velocity

Turning acceleration– acceleration for the turning velocity of both L&R axis.

Velocity L-Velocity of the left wheel during turning (different from straight velocity mentioned on motor constant).

Velocity R-Velocity of the right wheel during turning (different from straight velocity mentioned on motor constant).

Both two velocities mentioned above determine the arc of turning, so in the turning, select the appropriate velocity of L&R axis is very important. Normally, outer and inner wheel velocity ratio at the range of 2~4, usually the ratio is 3.

5.5. Front Correction

After the micromouse sensed a front wall in front of it, it will have a delay before performing the turn. This is used to enhance the stability of the mouse, as shown in Fig. 13.

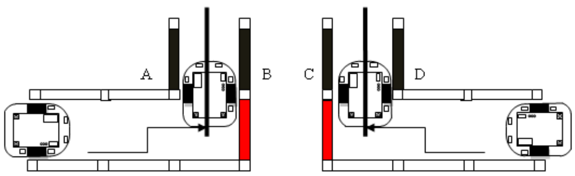


Fig. 13. Desired front wall correction.

When micromouse make a left 90° turn, in order to accurately complete the turn action, the front sensor of micromouse will make a front L90 correction, after the correction, the micromouse should lie in between the wall highlighted in black as shown in Fig. 13. If micromouse doesn’t lie in the desired position after the correction, can change the value according the following rules:

1) Increase the value if the micromouse is nearer to point A.

2) Decrease the value if the micromouse is nearer to point B.

When micromouse make a right 90° turn, in order to accurately complete the turn action, the front sensor of micromouse will make a front R90 correction, after the correction, the micromouse should lie in between the wall highlighted in black as shown in Fig. 13. If micromouse doesn’t lie the desired position after the correction, can change the value according the following rules:

1) Increase the value if the micromouse is nearer to point D.

2) Decrease the value if the micromouse is nearer to point C.

5.6. After Turn Correction

After the micromouse do a left/right turn during searching, if there is a wall in front of it, it will use the wall to compensate the distance offset error. This is used to enhance the stability of the mouse.

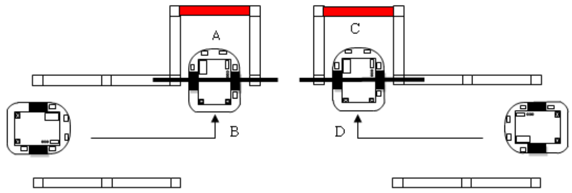


Fig. 14. Desired after turn correction.

When micromouse make a left 90° turn, in order to accurately complete the turn action, the front sensor of micromouse will make a L90 after turn correction, after the correction, the micromouse should lie in between the wall highlighted in black as shown in Fig. 14. If micromouse doesn’t lie in the desired position after the correction, can change the value according the following rules:

1) Increase the value if the micromouse is nearer to point B.

2) Decrease the value if the micromouse is nearer to point A.

When micromouse make a right 90° turn, in order to accurately complete the turn action, the front sensor of micromouse will make a R90 after turn correction, after the correction, the micromouse should lie in between the wall highlighted in black as shown in Fig. 14. If micromouse doesn’t lie in the desired position after the correction, can change the value according the following rules:

1) Increase the value if the micromouse is nearer to point D.

2) Decrease the value if the micromouse is nearer to point C.

6. Calibration of Micromouse Static Turn

This section for basic turning of micromouse, which static run (180, L90, R90), in order to ensure accurate turning of micromouse, using the following parameters to help the turning: turning angle and turning velocity. Static turn calibration software designed by VB as shown in Fig. 15.

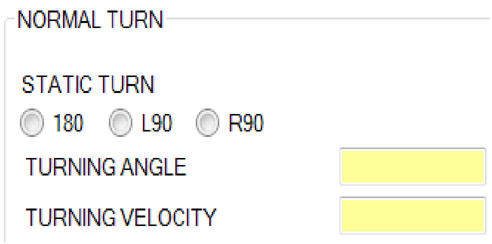


Fig. 15. Static turn calibration software by VB.

Can be seen from Fig. 15, static turning calibration is not so complex as smooth turning. It's calibration parameters only includes turning angle and turning velocity.

Turning angle - values used to determine how much the micromouse must turn in order to perform a static turn (180, L90, R90).

A static turn 180 and its calibration as shown in Fig. 16. As can be seen from the figure, if the parameter is selected wrongly, micromouse cannot do a 180 degree turn. If micromouse doesn't lie in the desired position after the turn, can change the value according the following rules:

1) Increase the value if the micromouse is nearer to point A.

2) Decrease the value if the micromouse is nearer to point B.

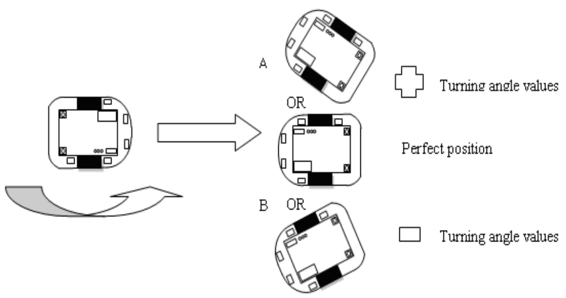


Fig. 16. Static 180 calibration.

A static left 90° turn and its calibration as shown in Fig. 17. As can be seen from the figure, if the parameter is selected wrongly, micromouse cannot do a 90° turn. If micromouse doesn't lie in the desired position after the turn, can change the value according the following rules:

1) Increase the value if the micromouse is nearer to point A.

2) Decrease the value if the micromouse is nearer to point B.

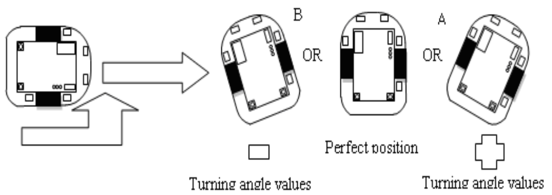


Fig. 17. Static L90 calibration.

A static right 90° turn and its calibration as shown in Fig. 18. As can be seen from the figure, if the parameter is selected wrongly, micromouse cannot do a 90° turn. If micromouse doesn't lie in the desired position after the turn, can change the value according the following rules:

1) Increase the value if the micromouse is nearer to point A.

2) Decrease the value if the micromouse is nearer to point B.

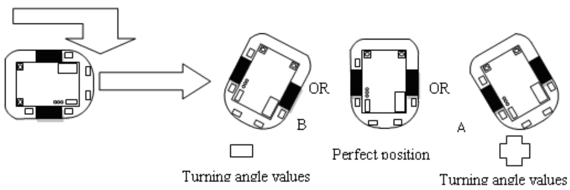


Fig. 18. Static R90 calibration.

In this calibration section, turn velocity and turning acceleration are other key parameters. turn velocity - the velocity for performing the static turn, which determine the whole rotation time. Turning acceleration-the acceleration for performing the static turn, which determines the rotational stability.

7. Calibration of Micromouse Parking

When the micromouse reaches a dead end, if it detects some parking position error, it will do the parking correction by itself, as shown in Fig. 19. In parking calibration, there two key parameters: PC_Vel: The velocity of doing the parking correction; PC_Gain: The strength of doing the parking correction.

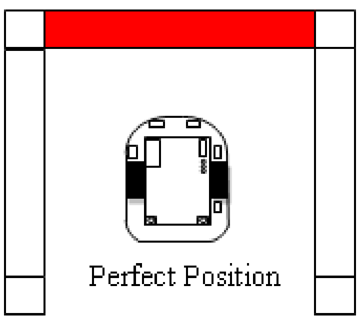


Fig. 19. Perfect parking position.

8. Conclusions

To meet the special requirements of high speed micromouse turning in complex maze, dual core micromouse based on ARM9 and FPGA is proposed, In order to increase the system stability and decrease response time, stepper motor is substituted by DC motor. In order to improve micromouse competition results, reduce run time of micromouse, fast turning principle of micromouse is analyzed, different parameters were used to control its servo action, and other parameters to adjust its attitude, which can ensure the rapidity, accuracy and good stability of micromouse in high speed turning.

Acknowledgement

It is a project supported by basic research programs of Suzhou science and Technology Department - industrial application part (SYG201327) and 2012 Innovation Project of JiangSu Province.

References

[1]. Su Juing-Huei, Lee Chyi-Shyong, Huang Hsin-Hsiung, etc., A micromouse kit for teaching autonomous mobile robots, *International Journal of Electrical Engineering Education*, 48, 2, 2011, pp. 188-201.

[2]. Mishra Swati, Bande Pankaj, Maze Solving Algorithms for Micro Mouse, in *Proceedings of the IEEE International Conference on Signal Image Technology and Internet Based Systems*, 2008, pp. 11.

[3]. Kiblcr S. G., Ilauer A. E., Giesscl D. S., etc., IEEE Micromouse for mechatronics research and education, in *Proceedings of the IEEE International Conference on Mechalronics*, 2011, pp. 887-892.

[4]. Po-Jen Cheng, Chin-Hsing Cheng, Chun-Cheng Tung, Design of DC motor's torque control using DSP, *Journal of Information and Optimization Sciences*, 30, 6, 2009, pp. 1197-1207.

[5]. Zhang Haoming, Peh Lian Soon, Wang Yinghai, Two axes micromouse PWM generator based on FPGA, *Applied Mechanics and Materials*, 496-500, pp. 1674-1680.

[6]. Zhang Haoming, Peh Lian Soon, Wang Yinghai, Design of a Edubot - Micromouse based on MCU+FPGA, *Applied Mechanics and Materials*, 496-500, pp. 1664-1669.

[7]. Jiang Xiong, Ren Hua-Long, Ma Zhong-Li, Research on How ARM-based Micromouse Gets Out of Maze, *Modern Electronics Technique*, 08, 2011, pp. 14-16.

[8]. Tondr A. D., Hall Drew, The inception of chedda: a detailed design and analysis of micromouse,: *University of Nevada*, Las Vegas, 2004.

[9]. Zhou Yi-Min, Fu Jun-Hua, Speed control of micro-mouse based on fuzzy control, *Microcomputer Information*, 13, 2010, pp. 25-29.

[10]. Fu Xiu-Wei, Zhang Hua, Design of the MicroMouse hardware based on the ARM-M3, *Journal of Jilin Institute of Chemical Technology*, 01, 2012, pp. 47-49.

[11]. Wen Ru-Chun, Xu Ying, Wang Zu-Lin, Study and Application on Maze Path Planning Based on Improved Ant Colony Algorithm, *Journal of Jiangxi University of Science and Technology*, 02, 2010, pp. 26-28.

[12]. He Shao-Bo, Sun Ke-Hui, Design and Optimization of Micro-mouse Solving the Maze Algorithm Based on Central Method, *Computer Systems & Applications*, 09, 2012, pp. 79-82.

[13]. Chen Yu, Yang Ji-Min, Improvement and Search Algorithm of Micromouse Circuit, *Modern Electronics Technique*, 10, 2011, pp. 68-70.

[14]. Bo En, Yu Yuan Yuan, Design and Implementation of Micromouse Based on STM32, *Value Engineering*, 20, 2011, pp. 136-137.

[15]. Qiu Zhil, Li Ang, Zhang Wenjin, Design of Micromouse System based on ARM and FPGA, *Instrumentation Analysis Monitoring*, 2013, 02, pp. 88-91.