Sensors & Transducers, Vol. 170, Issue 5, May 2014, pp. 67-72

www.sensorsportal.com

Sensors & Transducers

http://www.sensorsportal.com

Auto-Navigation of Micromouse Based on Infrared Sensor

[1, 2] Zhang Haoming, [2] Wang Yinghai, [3] Peh Lian Soon
[1] Department of Electrical and Information Engineering, Tongling University,
Anhui Province, 214000, China
[2] School of Electrical and Information Engineering, SIPIVT, Jiangsu Province, 215123, China
[3] Shining Sunshines Global PTE LTD, Singapore
[1] Tel.: 13013801792
[1] E-mail: zhm20060616@163.com

Received: 3 March 2014 /Accepted: 30 April 2014 /Published: 31 May 2014

Abstract: Micromouse is an intelligent robot that is designed to search a shortest path to the destination in a unknown maze, in order to make the mouse can memory the right complex maze information automatically after searching and dashing, different parameters decided by infrared sensors were used to record micromouse's position in the maze, also is used as the reference to realize micromouse position compensation, which can ensure the rapidity, accuracy and good stability of micromouse in high speed exploration and dashing. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Micromouse, Embedded technology, Maze, Auto-navigation, Exploration, Dashing.

1. Introduction

Micromouse championship is an international robotics competition, which is an event where a small robot micromouse solves a 16x16 maze or 32x32 maze according to different competition conditions [1-4], which is completely autonomous that is able to find their way from a predetermined starting position to the central area of the maze unaided, the maze and the micromouse as shown in Fig. 1.

The robot will need to keep track of where it is, discover walls as it explores, store the maze and detect whether it has reached the goal. After it reached the goal, the robot will typically perform additional searches of the maze until it has found an optimal route from the start to the center. Once the optimal route has been found, the robot will run that route in the shortest possible time [6-8].

Now the events are held worldwide, and are most popular in the UK, USA, Japan, Singapore, India, South Korea and China.

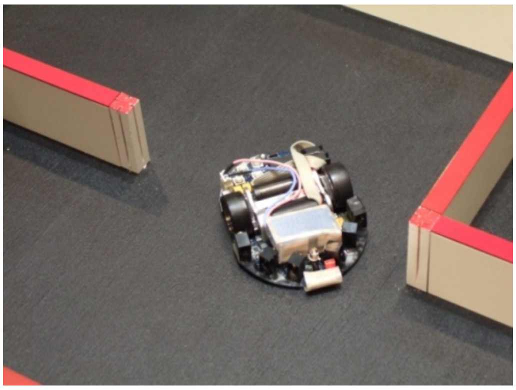

Fig. 1. Maze and micromouse.

Microumouse system comprises two main parts which are the electronic and mechanical systems. For the electronic system, it is mainly controlled by a MCU. All of the calculations and decision makings will be executed by the MCU according to the value feedback by six pairs of infrared sensors for the obstacle detection [9-11].

2. Infrared Sensing Module

Infrared sensor is an electronic device that emits infrared radiation and by detecting the reflection of the radiation through the receiver. Infrared sensors are easy to design and assemble because the components required are easy to come by [12-14].

The placement of the transmitter and receiver of the infrared sensor for the micromouse as shown in the Fig. 2.

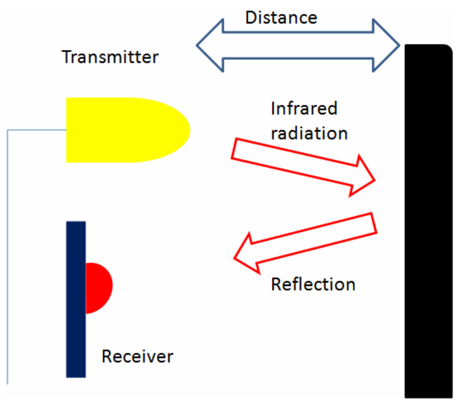

Fig. 2. Placement of infrared sensor.

The red color arrows show the infrared radiation and after it hit the wall, it will reflect back to the receiver. By measuring the voltage produced by the receiver, the distance between the sensors can be approximated. By placing a resistor between the power supply and the infrared emitter can manipulate the intensity of the emitter so that the receiver would not get saturated easily by detecting large amount of infrared radiation.

The robot has 6 pairs of infrared transmitters and receivers, so that it can detect the walls for maze solving and auto-navigation. Its principle is illustrated in Fig. 3.

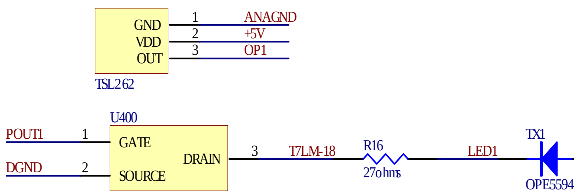

Fig. 3. Infrared sensor transmitting and receiving circuit.

In the circuit, OPE5594 emits the infrared light, change the value of R16 can change the intensity of the reflected light,TSL262 accepts the reflected infrared light and convert it, then input the value to MCU.

The transmitters are directly controlled by microcontroller's I/O ports and the system gets the readings from receivers by using the built-in analog-to-digital converter (ADC).

In order to reduce the unwanted noise, the author tries to get two readings for one channel per update, one is sampled when the transmitter is turned on ($V_{Transmitter\_On}$) and another one is sampled when the infrared-LED is off ($V_{Transmitter\_Off}$), then the difference between these two readings will be the final output of the sensor (V_{Sensor}). We establish V_{Sensor} to be

$$V_{Transmitter\_On} - V_{Transmitter\_Off} = V_{Sensor} \qquad (1)$$

This method is to minus the reading of noise from surrounding which is the $V_{Transmitter\_Off}$, so that the effect of varying surrounding lighting condition will be minimized.

The system has to turn on the next transmitter LED from channel 1 to 6 right after one analogue-to-digital conversion is finished as the LEDs will have to take some time before they are fully on and the LEDs must be fully on before the analogue-to-digital conversion is started so that the system is able to take the correct readings.

The LEDs will also have to take some to turn off completely, but the next conversion will be started in very short time. There will be some unwanted reflections and they will be the noise. The author has changed the sensing sequence from 1-2-3-4-5-6 to 1-4-2-5-3-6, so that the unwanted reflections will not affect the channel that is currently sampling, as shown in Fig. 4.

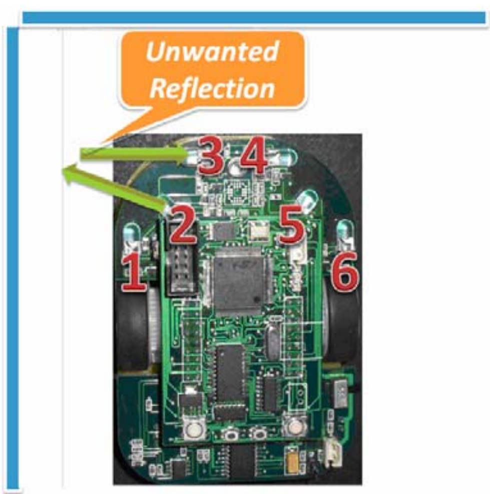

Fig. 4. Sensing module.

The original output of ADC is 12-bit long, and the MCU used in the robot is an 8-bit or 12-bit microcontroller, if an 8 bit MCU, the lower 4 bits of data will be ignored by shifting the output data into an 8-bit data, as shown in Fig. 5.

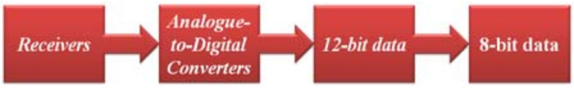

Fig. 5. 8-bit ADC output principle.

Then the sensor values will vary from 0x00 to 0xFF and this can also help to reduce the undesired noise which is always smaller and appears in lower 4 bits.

3. Navigation for Straight Paths

The robot has to remain in the centre of the cells when travelling in the maze, so that the auto-navigation must be integrated into the system depends on the infrared sensors in the system.

3.1. Side-wall Navigation

The robot can determine its position by using the feedback from the six pairs of infrared sensors.

The robot will firstly detect how many side-walls are there. If there is only one wall on either left or right side, the robot will check whether it is near to the side-wall or far away from that and then perform the suitable navigation action. And if there are two side-walls, then the robot can easily check whether the left side-sensor value is larger than the user defined threshold or the right side-sensor value is large than the threshold, then navigate to right side if the left side-sensor feedbacks a larger value and move to the left side if the right side-sensor gives a larger feedback. In this paper use variables V23 and V24 store value of side-sensor, which is illustrated in Fig. 6. V24H stands reference value of Sensor 2 when mouse in the center of a cell, V24L stands reference value of Sensor 5 when mouse in the center of a cell, V23H stands reference value of Sensor 3 when mouse in the center of a cell, V23L stands reference value of Sensor 4 when mouse in the center of a cell.

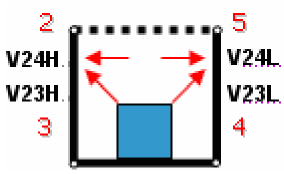

Fig. 6. Side-sensor for navigation reference.

For the single wall cases, if the side-sensor feedbacks a smaller value compare to the user defined centre reference which is calibrated by putting the robot in the centre position, then it means that the robot is far away from the side-wall.

3.2. Distance Offset Compensation

The robot can compensate for the offset which is accumulated when it is traveling along a long straight path by replacing the calculated distance with another user defined distance after it detects a high-to-low transition which means it passes through a cell with side walls to a cell with only one side wall or no side wall. In this paper use variables V25H, V25L, V26H, V26L, V27H, V27L, V28H and V28L store value of sensor transition states.

Wall references of sensors in transition straight high are illustrated in Fig. 7. V25H stands reference value of Sensor 2 when mouse in transition high, V25L stands reference value of Sensor 5 when mouse in transition high, V27H stands reference value of Sensor 3 when mouse in transition high, V27L stands reference value of Sensor 4 when mouse in transition high.

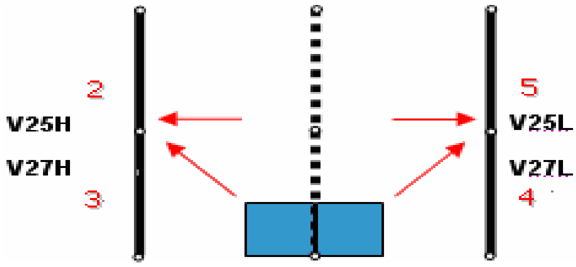

Fig. 7. Transition straight high reference.

Wall references of sensors in transition straight low are illustrated in Fig. 8. V28H stands reference value of Sensor 3 when mouse in transition low, V28L stands reference value of Sensor 4 when mouse in transition low, V26H stands reference value of Sensor 2 when mouse in transition low, V26L stands reference value of Sensor 5 when mouse in transition low.

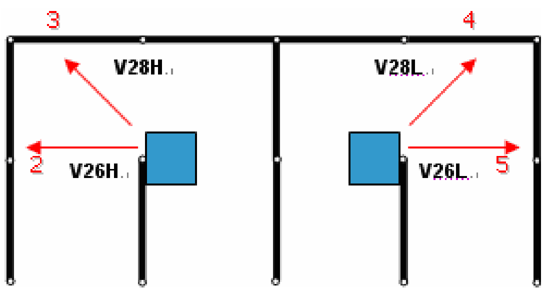

Fig. 8. Transition straight low reference.

The distance offset compensation routine will keep detecting the side-walls' condition while the robot is travelling along a straight path. If the robot cannot sense the wall after a wall has been detected before, that means there is a 'high-to-low' transition found in the straight path, the robot is then able to find out its actual position since the sensors' positions are fixed on the robot. Then the compensation routine will replace the distance set-point by the user defined value instead of using the distance that is calculated which may be no longer accurate as the slippage may be occurred when the robot is traveling along the long straight path. When the robot finishes traveling the compensated distance, it will reach the new cell nicely.

3.3. Diagonal Path Navigation

After the robot finishes searching the maze and if there is a path containing consecutive 90-to-90 turns, then it can shorten the travelled distance for saving the travelling time by performing a diagonally straight moving.

But there is no any parallel side-wall for the robot to execute the normal straight path navigation, so that another auto-navigation method has to be implemented in order to help the robot from the collision between the robot and walls. As there is no any wall parallel to the robot when it is travelling in a diagonal straight path, so that the robot will not be able to determine the position by using side or diagonal sensors.

The robot can still use the two front sensors to determine whether it is heading in right direction. Fig. 9 shows micromouse is in wrong heading when in diagonal path. When it is in the wrong direction, the two front sensors will sense object which means the two sensor readings should be remain high. When one of them feedbacks a larger value compared to the user defined threshold, that means the robot is heading in wrong direction and it may crash with the obstacle if it keeps travelling in that direction.

Fig. 10 shows micromouse is in right heading when in diagonal path. When it is in the right direction, the two front sensors will not sense any object which means the two sensor readings should be remain low.

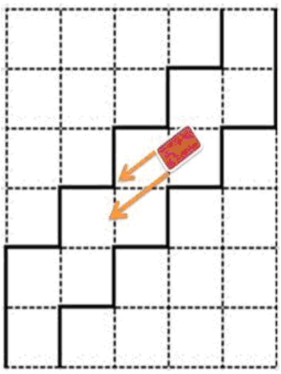

Fig. 9. Wrong heading in diagonal path.

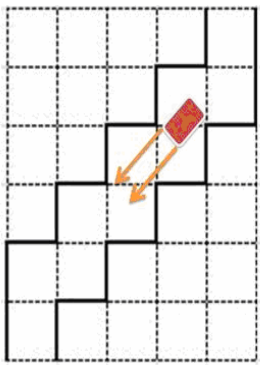

Fig. 10. Right heading in diagonal path.

The diagonal navigation function will keep checking the feedback of two front sensors. If both of them feedback the larger value compare to the user defined thresholds, then this function will be end because the robot may probably come to the end of the diagonal path and the two front sensors sense a real front wall, so no exact conclusion can be made at that time and it should not navigate the robot in this case. The diagonal navigation is similar to the side-wall navigation which means that the robot also uses two sensors that are on different sides to detect the obstacle and navigate itself. Instead of using two side-sensors to detect the distance between the robot and side-walls, the diagonal navigation function uses two front sensors to detect the error in heading direction. Once one of them feedbacks an abnormal value and the other sensor value remains smaller, then the navigation action will be taken.

4. Special Navigation

4.1. Front-wall Compensation

The robot can determine the distance between itself and the front wall before performing a left or right turn, so that it can try to compensate for the distance error accumulated while travelling in a long straight path in order to turn in the correct position.

If the robot senses a front-wall before it turns, then it will not travel for the calculated distance and will update the travelling distance by comparing the front sensor values with the user-defined values, as shown in Fig. 11.

4.2. Parking by Front-wall Correction

The robot will stop in the maze when its route is blocked by a wall or it is solving the maze for a long time. The robot may not be able to navigate back to the centre of cell due to the immediate stops. A front-wall correction for parking is developed to compensate for the heading errors, which is illustrated in Fig. 12. V19H stands reference value of Sensor 1 when micromouse need home parking, V19L stands reference value of Sensor 6 when micromouse need home parking.

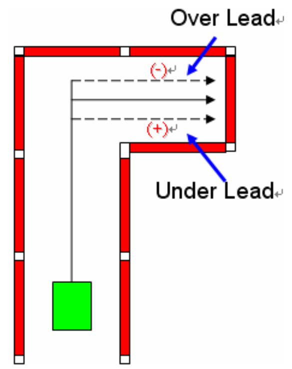

Fig. 11. Front wall compensation reference.

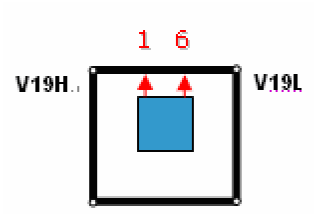

Fig. 12. Home parking reference.

The robot will start to slow down after it has decided to stop. The loop will keep running until both of the motors are stopped. The robot will stop the motors by checking the respective front sensors. If the sensor value is larger than or equal to the user-defined value, then it will stop the respective motor. The reason why it stops the motors in this way is to park the robot in right heading direction.

4.3. Dead end Compensation

When the robot is exploring around the maze, it may turn into a dead end where the cell has three surrounding walls, as shown in Fig. 13, using different variables V33H and V33L to judge dead end. V33L stands reference value of Sensor 1 adds Sensor 6 when the front maze maybe a dead end, V33H stands reference value of Sensor 1 adds Sensor 6 when the front maze is a dead end.

After dead end is conformed, the robot can only make a stationary 180° turn to continue its exploration, which is illustrated in Fig. 14.

When the robot is travelling around the maze, it may slip out of the centre of cell and it has to stop immediately when there is a wall blocking in the way it plans. In this case, the parking by front-wall correction can only compensate for the heading error, but it may be still not in the centre of the cell.

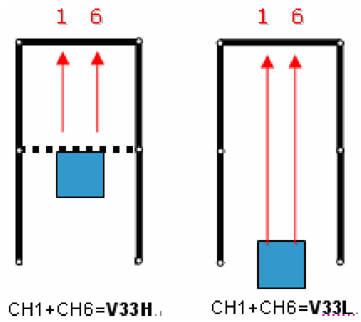

Fig. 13. Dead end reference.

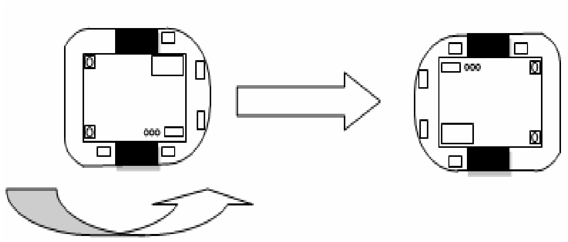

Fig. 14. Stationary 180° turn.

Dead End Compensation is developed to correct the parking position, so that the robot is able to start from the centre of cell and the probability of the collision between the robot and walls will be smaller.

The dead end compensation will be executed whenever the robot is stopped and it will try to detect whether it is in a dead end by comparing the sensor values with the wall references. If the robot is still in the centre of cell, then the dead end compensation will be bypassed even it is in a dead end.

Robot will perform a short backward move for the right motor followed by a short backward move for the left motor. These two short moves will help the robot to navigate to the right side, but it will be far away from the front-wall due to these two backward moves. So it will perform a forward move for both of the motors and they will stop independently by checking their own respective front sensors.

The main loop of dead end compensation will also check whether it has performed the moving action for more than three times. After the three moving actions, it will stop the dead end compensation.

5. Conclusion

Six pairs of infrared sensors were used for the obstacle detection of micromouse in maze. Different parameters decided by infrared sensors were used to record micromouse's position in the maze, also to realize micromouse position compensation, which ensures the rapidity, accuracy and good stability of micromouse in high-speed exploration and dashing in complex maze.

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*, 11, 2008.

[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]. Jiang Xiong, Ren Hua-Long, Ma Zhong-Li, Research on How ARM-based Micromouse Gets Out of Maze, *Modern Electronics Technique*, 8, 2011, pp. 14-16.

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

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

[8]. Fu Xiu-Wei, Zhang Hua, Design of the Micromouse hardware based on the ARM-M3, *Journal of Jilin Institute of Chemical Technology*, 1, 2012, pp. 47-49.

[9]. 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*, 2, 2010, pp. 26-28.

[10]. He Shao-Bo, Sun Ke-Hui, Design and Optimization of Micro-mouse Solving the Maze Algorithm Based on Central Method, *Computer Systems & Applications*, 9, 2012, pp. 79-82.

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

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

[13]. Qiu Zhil, Li Ang, Zhang Wenjin, Design of Micromouse System based on ARM and FPGA, *Instrumentation Analysis Monitoring*, 2, 2013, pp. 88-91.

[14]. Barun Ray, Chaudhuri, On the determination of the emission wavelength of an infrared LED with common laboratory instruments, *European Journal of Physics*, 32, 4, 2011, pp. 935-945.