Proceedings of 2016 International Conference on Advanced Robotics and Intelligent Systems
Taipei Nangang Exhibition Center, Taipei, Taiwan, 31 August~2 September, 2016

The development of a half-size micromouse and its application in mobile robot education

Juing-Huei Su, Xin-Han Cai, Chyi-Shyong Lee, and Chao-Wei Chen
Department of Electronic Engineering, Lunghwa University of Science and Technology
Taoyuan, 33306, Taiwan
E-mail: suhu@mail.lhu.edu.tw

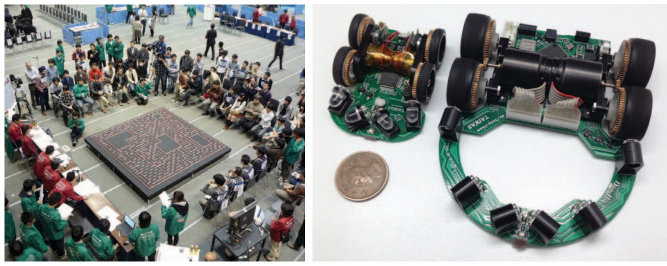
***Abstract*—The development of a half-size micromouse for international contests and mobile robot education is presented in this paper. It is used in a project-oriented course for senior undergraduate and graduate students about mobile robots, because the related contest is fun for students to explore many disciplines of knowledge and skills. To make a micomouse run in an unknown maze, students need to integrate the skills of designing 1) mechanical parts, 2) its printed circuit board, 3) sensor calibration procedures, 4) digital motion control algorithms, 5) path planning algorithms, and 6) maze solvers. To win some prize in a contest also requires students to devise some advanced new algorithms for a micromouse to find the best route and run faster. Therefore, improved algorithms about 1) a simple and efficient time-based diagonal maze solver, and 2) position and velocity resolution enhancement with accelerometers are also presented in this paper.**

Keywords—half-size micromouse; robot education;

I. INTRODUCTION

Robots are playing an increasingly important role in exploration and our daily life, especially when the exploration rover, Curiosity, landed on Mars [1], and iRobot introduced the vacuum-cleaning robot [2]. Understanding the construction and control of mobile robots has therefore become a necessity for many electrical and electronic engineers. Unfortunately, the robot design draws on many areas of knowledge, such as mechanics and electronics, automatic control theory, and software programming of microcontrollers [3-4]. Nevertheless, students are willing to do tedious research work to solve practical problems when these are related to an interesting, competitive contest [5-7]. Winning one or two awards in a competition not only gives students a sense of accomplishment, but also gives their schools pride and visibility. This is an important factor for technology-oriented university students in Taiwan, who tend to have low learning achievements in traditional theory-oriented lecture courses. The original Micromouse contest, in which autonomous robots compete in terms of speed and intelligence, were started more than 30 years ago. Because the robots' performance is still improving, these contests are still very popular with engineering students in the U.K., US, Singapore, Japan [8], and Taiwan [9]. To challenge the contestants even more, the rules were changed in Japan in 2010 to adopt a larger maze (32×32), an arbitrary goal area, and smaller maze cell dimensions (Fig. 1).

Micromouse is basically an autonomous robot that has to search from the start cell to the goal in an unknown maze for the shortest path. The goal is fixed in the center for classic contests (16×16), and can be any area in the maze for the new half-size contests. The micromouse has to run as fast as possible from the start cell to the goal area to win the contest.



(a) (b)

Fig. 1. (a) Half-size micromouse contest in Japan, (b) Comparisons of classic (right) and half-size (left) micromouse

Although the objective seems simple, the contestants have to devise an efficient maze solver such that the shortest path can be found within a time limit. Because of the fast development of microcontrollers and micro-electro-mechanical-system (MEMS) sensors, the micromouse can be built to be small that it can run diagonally in the maze. This poses a new problem to maze solvers whether or not they can find the best diagonal route from the start cell to the goal, and take into account the motion capabilities of the micromouse. Furthermore, the challenge of small torques that toy dc motor can provide forces half-size micromouse contestants to use magnetic encoder or lower resolution encoders [10-11] such that the micromouse can be lighter to obtain larger acceleraltions (Fig. 2). Therefore, this paper also proposes 1) a new simple and efficient diagonal maze solver for micromouse contests, and 2) resolution enhancement algorithms for positions and velocities with the help of accelerometers, to tackle these problems. These experiences also help make the project oriented course more challenging.

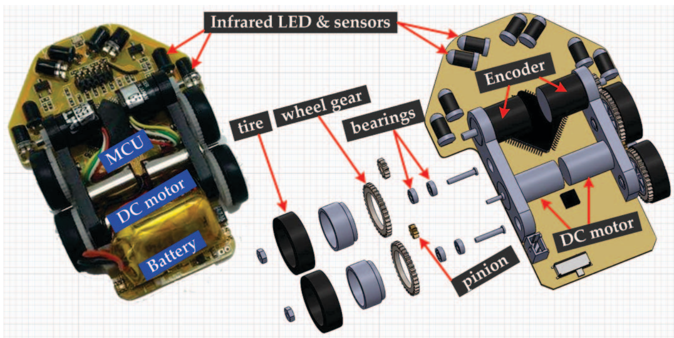


Fig. 2. A half-size micromouse devised in a project oriented course

978-1-5090-5102-1/16/$31.00 ©2016 IEEE

II. The Development of a Half-size Micromouse

A. *Hardware circuit and mechanical structure*

The half-size micromouse in Fig. 2 is a 4-wheel differential wheeled robot. The reason for 4-wheel differential driven configuration and not 2-wheel structures is the higher coefficient of friction it can provide during acceleration and deceleration, although extra torque is needed in making turns. All the motor mount and wheel configurations shown in Fig. 2 are custom designed by using the 3D computer aided tool, SolidWorks. Bearings, tires, washers, and screws are bought from local distributors. The other tiny mechanical parts are fabricated with a desktop 3D printer [12] or a computer numerical control (CNC) machine [13]. The half-size micromouse controls 4 pairs of infrared light emitting diodes (LEDs) in 4 directions, and detects the intensity of the reflected light with optical sensors to determine the maze wall information. These intensity values of reflected light are also used to correct the orientation, lateral and longitudinal moving errors of the micromouse in the maze by controlling the speeds of two DC motors. Magnetic encoders and a gyro sensor are also built into the mouse, such that the moving distance and rotation angle can be accurately calculated. A powerful 32-bit microcontroller with floating point operation unit, RX-62T, freely supported by the local Renesas distributor is used to read sample values of optical sensors, encoder pulses, and the gyro output, to control the motions, to plan a moving path, and to search for the shortest route from the start cell to the goal area with a maze solving algorithm.

The hardware circuits and components used on the half-size micromouse are summarized in Table I. To make the half-size micromouse light and small, encoder signals and LEDs are used for user interactions and operation mode settings. Therefore, neither button nor matrix LED display is necessary in hardware circuit design. Students can also use the universal asynchronous receiver/transmitter (UART) serial bus on the microcontroller and the universal serial bus (USB) on the personal computer to collect the data about motion control, peripheral control, and maze-solving algorithms for debugging purposes.

TABLE I. The Biodata of the Half-size Micromosue

Biodata	Half-size micromosue
Weight, Length, Width, Height	20g, 53.4 × 38.0 × 16.5 (mm)
Gears (Spur:Pinion)	M0.3 (48:12)
Magnetic encoder, Gyro	AS5040, LY3200ALH
Processor	Renesas, RX631
Turn and Straight Speed, Acceleration	0.6 m/s, 1.0 m/s, 5 m/s^2
Motor (driver)	MK06-4.5 (DRV8836DSSR)
IR transmitter (receiver)	SFH4550(RPM-012PB)
Battery (Lithium Polymer)	70mAh 1P1S

B. *An interrupt driven real-time control firmware*

There are three parts in the motion control of the half-size micromouse, 1) position and center velocity control, 2) angle and angular velocity control, and 3) position and attitude correction. Fig. 3 shows the structure of how these three parts are implemented, where v_{Cr}, ω_{Cr}, p_{Cr}, θ_{Cr}, $\hat{v}_C$, $\hat{\omega}_C$, $\hat{p}_C$, $\hat{\theta}_C$, v_R, v_L, θ_R, θ_L, W stand for the reference center and angular velocities, reference center and angular positions, estimated center and angular velocities, estimated center and angular positions, right and left motor velocities, right and left motor angular positions, and the width between wheels, respectively.

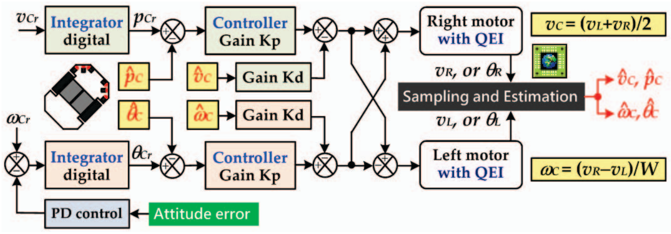


Fig. 3. Motion control block diagram for the devised half-size micromouse

To implement these motion control algorithms with a fixed period of sampling time, an interrupt-driven firmware structure is used in this half-size micromouse. The sampling intervals should be short to make the micromouse react fast enough to environment changes, which is important when the micromoue is running fast. Key factors to influence the code execution efficiency, such as fixed-point mathematics, table-lookup skills for trigonometric functions, and programming skills would be the main learning issues for students in this part. The other important part of the firmware is to solve the maze and find the best route from the start cell to the goal area. The maze solver would be triggered by the interrupt-driven subroutine at an appropriate place of a cell after new maze wall information is collected with optical sensors. The control flow of the firmware is illustrated in Fig. 4.

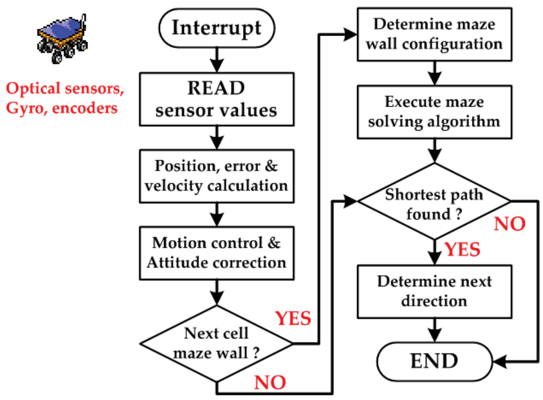


Fig. 4. The interrupt driven firmware structure of motion control

C. *Observer based sensor fusion algorithm [14]*

Assume that a_{xc} and p_c stand for the acceleration in forward direction of a mobile robot, and half of the sum of the encoder counts of the two motors, respectively. The following dynamical equations describe the relationship of these variables.

$$\begin{bmatrix} \dot{p}_C \\ v_C \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_C \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_{xc}, \tag{1a}$$

$$p_C = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_C \\ v_C \end{bmatrix}. \tag{1b}$$

Therefore, the idea of state observer can be applied to reconstruct the state variables, by using the acceleration signal a_C, and the following dynamical equations

$$\begin{bmatrix}\dot{\hat{p}}_C\\ \dot{\hat{v}}_C\end{bmatrix}=\begin{bmatrix}0&1\\0&0\end{bmatrix}\begin{bmatrix}\hat{p}_C\\ \hat{v}_C\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}a_{xc}+\begin{bmatrix}k_v\\k_p\end{bmatrix}(p_C-\hat{p}_C), \tag{2a}$$

$$\hat{p}_C=\begin{bmatrix}1&0\end{bmatrix}\begin{bmatrix}\hat{p}_C\\ \hat{v}_C\end{bmatrix}. \tag{2b}$$

where $\hat{p}_c$, $\hat{v}_c$ stand for the estimates of the encoder counts p_C and center velocity v_C, respectively. The error dynamics could be obtained by subtracting (2a) from (1a)

$$\begin{bmatrix}\dot{e}_p\\ \dot{e}_v\end{bmatrix}=\begin{bmatrix}-k_v&1\\-k_p&0\end{bmatrix}\begin{bmatrix}e_p\\e_v\end{bmatrix}, \tag{3}$$

where $e_p=p_C-\hat{p}_C$, $e_v=v_C-\hat{v}_C$.

The control gains k_p, and k_v in (2a) could be used to make the errors in (3) between the true and estimate values approach to zero, if the roots of the characteristic equation of (3) in (4) lie in the left hand side of the complex plane.

$$s^2+k_v s+k_p=0. \tag{4}$$

Since the characteristic equation in (4) is a second order equation, the roots can be easily located and related to the idea of damping ratio and natural frequency by setting $k_p=\omega_n^2$, and $k_v=2\zeta\omega_n$. Therefore, the convergence speed of the error dynamics in (3) can be easily determined.

Rectangular method is used in experimental verification of this method. Fig. 5 shows the results in position and velocity control of a mobile robot. It can be seen that the linear velocity signal with observer-based sensor fusion method is far better than the one with encoder signals. The noise in the velocity signal also influences the PWM (pulse width modulation) command value for DC motors.

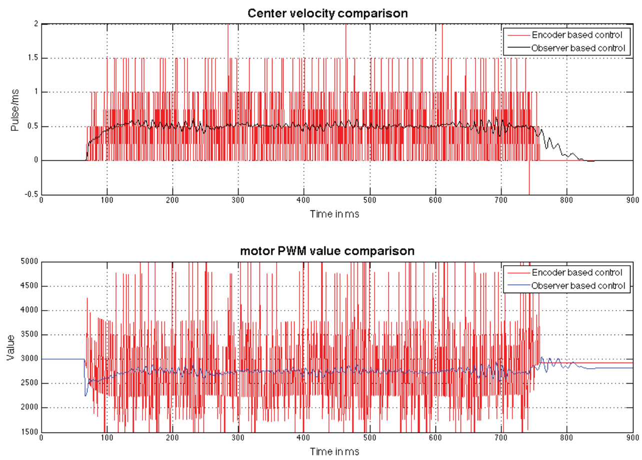


Fig. 5. Experimental results of the observer-based sensor funsion method

III. A TIME-BASED DIAGONAL MAZE-SOLVING ALGORITHM

There are many ways to solve the mazes found in the micromouse competitions. Traditionally, flooding algorithms [15] are very popular in finding the goal and best route of a given maze. The criteria of these solvers are mainly based on the distance of the route from the start cell to the goal area with right angle turns. Since the approach [15] does not take into account the motion capabilities for a micromouse to run in a diagonal, the flood values in each cell can be moved to borders between cells to remedy this problem shown in Fig. 6.

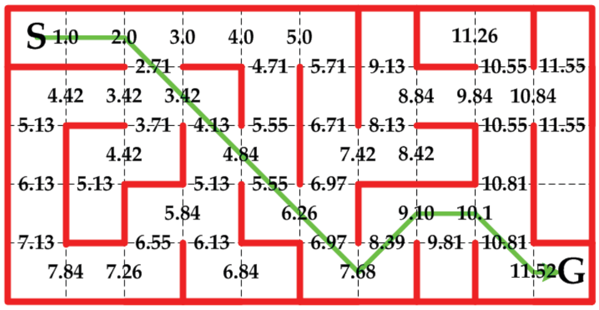


Fig. 6. A distance based maze solver with diagonal moving capabilities

The approach in Fig. 5 still does not take into account the differences of motions for a micromouse to run in a straight path or to make turns. This results in the development of the following time-based maze-solving algorithm [16]. This approach originates from the idea of considering the motion capabilities at straight and turning paths. If a micromouse can accelerate at straight paths and decelerate to a suitable speed for making turns, the maze solver should try to find a route with more straight paths. Since the flooding values are filled at the borders of maze cells to find diagonal routes, the time calculation for the flooding values would be based on the distance between these borders. Assume that the acceleration and deceleration of a micromouse are 10m/s^2 and -10m/s^2, respectively, and the turning speeds for 45, 90, 135, and 180 are all 1m/s. The top speed of a micromouse is basically limited by the motor specifications and how well the micromouse can align itself in normal or diagonal straight paths. It is assumed that the top speeds for a micromouse are 3.44 m/s for normal and 2.94 m/s for diagonal straight paths, after 3 cells of acceleration in this diagonal maze solver. Since the distance between classic maze cell borders is 18 cm, the time for a micromouse to run across a cell is 0.18s with the turning speed 1m/s. The necessary time for a micromouse to reach different borders in straight and diagonal straight paths can be calculated by using the following formulas:

$$v_{fn}^2=v_0^2+2(10)n\times0.18,\ n=1,\cdots,3, \tag{1a}$$

$$v_{fdm}^2=v_0^2+2(10)m\times\frac{0.18}{\sqrt{2}},\ m=1,\cdots,3, \tag{1b}$$

$$t_n=\frac{v_{fn}-v_0}{10},\ \Delta t_n=t_n-t_{n-1}, \tag{1c}$$

$$t_m=\frac{v_{fdm}-v_0}{10},\ \Delta t_m=t_m-t_{m-1}, \tag{1d}$$

where v_{fn}, v_{fdm}, v_0, n, m, t_n, and t_m stand for final velocities in straight and diagonal straight paths, initial velocity, number of normal and diagonal cells for acceleration, and the time needed for acceleration in straight and diagonal straight paths, respectively. Since the lowest speed in this case is the turn speed 1m/s, the initial velocity v_0 will be regarded as 1 in (1). Table II shows the time value in each cell for a micromouse to run in normal or diagonal straight paths. The normalized flood values for each cell in Table II are obtained by dividing Δt_n and Δt_m with a base value of 0.18s, which is the time value for a micromouse to run across a cell with a speed of 1m/s.

TABLE II. FLOOD VALUES FOR STRAIGHT PATH ACCELERATION

order of cells, *n*	1st	2nd	3rd	4th
Δt_n	0.114	0.072	0.058	0.052
Δt_m	0.088	0.059	0.047	0.043
Normalized Δt_n	0.63	0.40	0.32	0.29
Normalized Δt_m	0.49	0.33	0.26	0.24

To complete the time based flooding algorithm, the penalty flood values when making turns after acceleration for a number of cells should also be calculated. This is based on the observation that the speed for a micromouse should be decelerated to 1 m/s for making turns. Therefore, the following equations are used to find the penalty flood values

$$v_{ft}^2 = v_0^2 + 2(10)\frac{n}{2}\times 0.18,\ n = 1,\cdots,6, \tag{2a}$$

$$v_{ftd}^2 = v_0^2 + 2(10)\frac{m}{2}\times\frac{0.18}{\sqrt{2}},\ m = 1,\cdots,6, \tag{2b}$$

$$t_{n,t} = \frac{2(v_{ft}-v_0)}{10},\ \Delta t_{np} = t_{n,t} - t_n, \tag{2c}$$

$$t_{m,t} = \frac{2(v_{ftd}-v_0)}{10},\ \Delta t_{mp} = t_{m,t} - t_m, \tag{2d}$$

where v_{ft}, v_{ftd}, $t_{n,t}$, and $t_{m,t}$ stand for top velocities before deceleration in normal and diagonal straight paths, the necessary time for the micromouse to decelerate to the turn speed of 1m/s, respectively. The following table summarizes the normalized penalty values for a micromouse to decelerate to the base speed of 1m/s after acceleration.

TABLE III. NORMALIZED PENTALTY FLOOD VALUES FOR TURNS

order of cells, *n*	1st	2nd	3rd	4th	5th	6th
$t_{n,t}$	0.135	0.229	0.306	0.373	0.433	0.487
Δt_{np}	0.020	0.043	0.063	0.077	0.084	0.086
$t_{m,t}$	0.102	0.177	0.239	0.294	0.343	0.388
Δt_{mp}	0.013	0.030	0.045	0.056	0.062	0.064

Fig. 7 shows how Table II and III are used to fill flood values at borders of maze cells from the start. The 'S' at the lower left corner stands for the start cell. Since the micromouse first moves from center of the start cell in north direction, the initial flood value at the border of the start cell without maze wall is 0.5. The process is stopped after one of the borders in the goal area is filled with non-zero flood values (shown in Fig. 8). Three kinds of information are stored along the process of filling the flood values, 1) the flood value, 2) the direction from previous to current border, and 3) the number of cells that are in the same direction. Once the goal area is reached in the flood value filling process, the optimal diagonal path can be found according to the flood value information from the start cell to the goal area. Since the flood value filling process originates from the start cell, the optimal diagonal path would be constructed which begins with the largest flood value in the goal area. The next flood value or the border is chosen based on the following criterions:

1. If the number for the same direction k is greater than 1, the next border should be the one with number k-1;
2. If the number k at the first step is 1, the next border would be the adjacent one to the current with smallest flood values.

Fig. 8 shows the final result of the devised time-based diagonal maze solver when the 2011 all Japan classic micromouse contest maze is used. It is interesting that the chosen optimal path (red line) seems to be the longest one for distance based maze solvers. All the contestants except BengKiat Ng chose the other paths. This may be the reason why BengKiat Ng broke the 4-second limit (3.921) and won the championship.

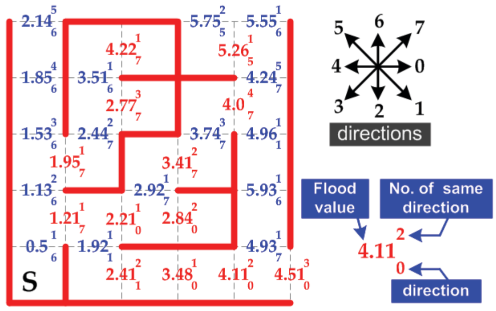


Fig. 7. The process of filling flood values with Table I and II

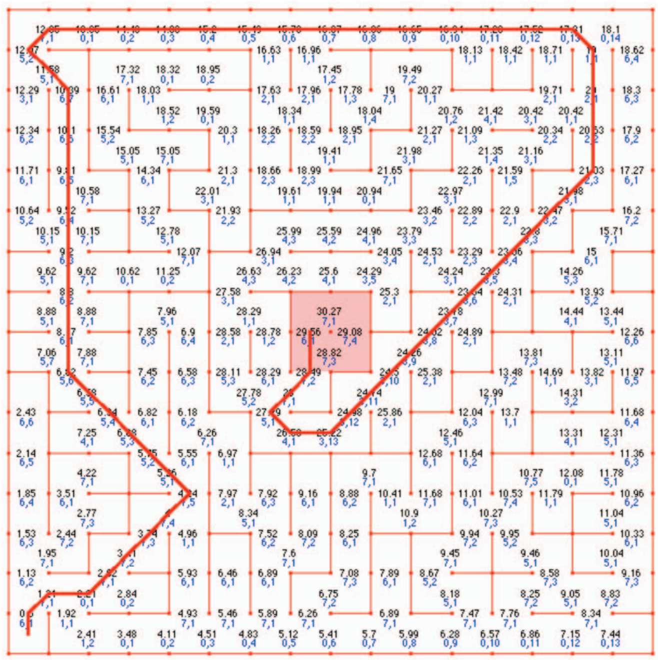


Fig. 8. The flood values by using the time-based diagonal maze solver

IV. THE MICROMOUSE IN MOBILE ROBOT EDUCATION

The implementation of a mobile robot requires multidisciplinary knowledge and skills, therefore, it is used in a contest oriented project for students in Department of Electronic Engineering, Lunghwa University of Science and Technology, Taiwan [17]. Most students initiate their mobile robot projects (classic micormouse, Robotracer, or line-maze robot [18]) out of interest in junior level, and get credits in a special topics course. This is because the project is fun and pushes them to explore continuously many disciplines of knowledge and skills. To get their credits at the end of the course of special topics, students have to pass an oral exam held by five teachers (not including their advisors), which tests the performance of their robots, and how well they had learned the implementation skills and the related theories. Most of the project students participated in local and even international contests to see how well their robot designs performed, and this is the other way to finish their project. The classic micromouse shown in Fig. 9 which won the first prize award both the 2015 All Japan and in 2016 APEC micromouse contests is such an example. The design and implementation of a half-size micormouse presented in this paper is a new trial for teachers and project students to compete with the other international contestants from this year on.

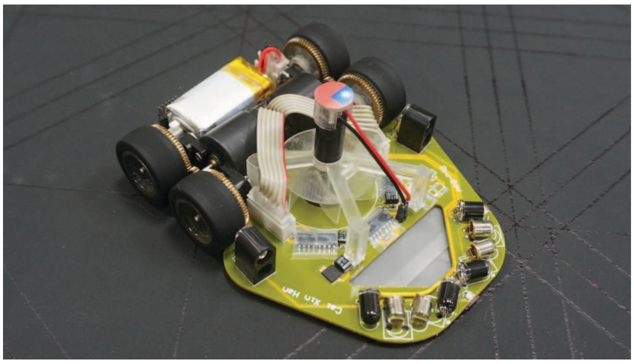


Fig. 9. The micromouse which won the championship of 2015 All Japan and 2016 APEC micromouse contests

Table IV lists the content that a project student who chooses to design his own micromouse could learn during the implementation process. The students would start with an educational micromouse shown in Fig. 10, which is also a project outcome in these years, to learn basic skills about microcontrollers and mobile robots.

TABLE IV. SKILLS AND THEORIES CONTAINED IN MICROMOUSE

Knowledge	Skills	Theories
Mechanics	✓3D software and printing;	-
Sensors	✓Calibrations; ✓Bandwidth effects;	✓Interpolation techniques for attitude correction;
Programming	✓Interrupt-driven firmware; ✓Peripheral control of microcontrollers;	✓Analysis of scheduling about multiple interrupt tasks;
Motion control	✓Acquire encoder signals; ✓PWM signals; ✓Behavior simulations;	✓Digital controller design; ✓Sensor fusion algorithms; ✓Path planning;
Maze solvers	✓Implementation of maze-solving algorithms;	✓Time-based flood values; ✓Goal searching strategies; ✓Criterions for finding the shortest path;

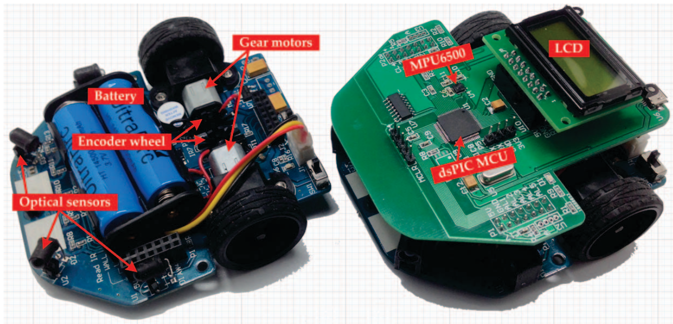


Fig. 10. The mobile robot for beginners in the projected oriented course

Once the project students are familiar with every part of the micromouse, they would start to design their own mobile robot. This process shown in Fig. 11 guides students step by step to explore deeply the knowledge and skills about mobile robots in

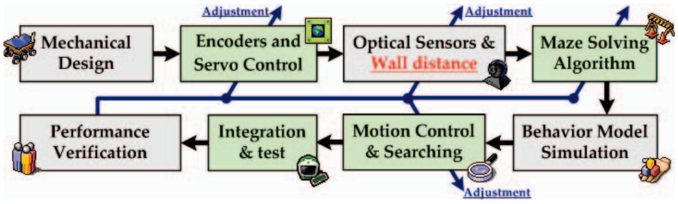


Fig. 11. The learning procedure for students in the project oriented course

V. CONCLUSIONS

A half-size micromouse is designed and implemented in this paper for international contests and mobile robot education. To make the micromouse more competitive, two new algorithms for improving the resolution of position and velocity estimations and a time based diagonal maze solver are also devised and presented in this paper. They are also parts of the project outcomes of students in Lunghwa University of Science and Technology. It is found that the micromouse is a very good topic for students to learn the implementation of mobile robots, because it contains multidisciplinary knowledge and skills. Students are willing to spend more time in these projects [17], even after getting their credits for the course. The incentives to win some prize, to discuss with international contestants, and to learn new knowledge and skills are major factors for the success of this contest oriented projects.

REFERENCES

[1] NASA's Curiosity Mars Rover, (Accessed 20 December, 2013.) Available from: http://www.nasa.gov/mission_pages/msl/.

[2] iRobot, Bedford, MA, "Vacuum cleaning robot—iRobot Roomba," (Accessed 20 December, 2013.) Available from: http://www.irobot.com.

[3] K. Nagai, "Learning while doing: Practical robotics education," *IEEE Robotics & Automation Magazine*, vol. 8, no. 2, pp. 39–43, Jun. 2001.

[4] K.-S. Hwang, W.-H. Hsiao, G.-T. Shing, and K.-J. Chen, "Rapid prototyping platform for robotics applications," *IEEE Transactions on Education*, vol.54, no. 2, pp. 236–246, May 2011.

[5] N. Chen, H. Chung, and Y. K. Kwon, "Integration of Micromouse project with undergraduate curriculum: A large-scale student participation approach," *IEEE Transactions on Education*, vol. 38, no. 2, pp. 136–144, May 1995.

[6] R. B. Murphy, "Competing for a robotics education," *IEEE Robotics & Automation Magazine*, vol. 8, no. 2, pp. 44–55, Jun. 2001.

[7] D. J. Pack, R. Avanzato, D. J. Ahlgren, and I. M. Verner, "Fire-fighting mobile robotics and interdisciplinary design-comparative perspectives,"

IEEE Transactions on Education, vol. 47, no. 3, pp. 369–376, Aug. 2004.

[8] New Technology Foundation, Tsukuba, Japan, “The 32nd All Japan Micromouse Contest,” (Accessed 24 December, 2011.) Available from: www.ntf.or.jp/mouse/micromouse2011/index_EN.html

[9] Facebook, Taiwan Micromouse and Intelligent Robot Contest, (Accessed 29 December, 2014.) Available from: https://www.facebook.com/TaiwanMIRContest

[10] Kojima Hirokazu, Velocity measurement with the Kojimouse7 encoder, (Accessed 8 January, 2015.) Available from: http://kojimousenote.blogspot.jp/2012/05/velocity-measurement-with-kojimouse7.html

[11] 竹本 裕太，加速度センサとロータリーエンコーダを組み合わせた速度計測法④：加速度センサによる問題解決へのアプローチ, (Accessed 8 January, 2015.) Available from: http://matsui-mouse.blogspot.jp/2014/03/blog-post_26.html

[12] Stratasys, Objet30 Pro- Precision desktop 3D printing, (Accessed 30 December, 2014.) Available from: http://www.stratasys.com/3d-printers/design-series/objet30-pro.

[13] Roland, MDX-40A - 3D milling machines, (Accessed 30 December, 2014.) Available from: http://www.rolanddg.com/product/3d/3d/mdx-40a/index.html.

[14] C.T. Chen, Linear System Theory and Design, 3rd edition, Oxford university press, 1999.

[15] Society of Robots, “Micromouse – Maze Solver – Theory,” (Accessed 12 July, 2013.) Available from: http://www.societyofrobots.com/member_tutorials/node/94

[16] J.-H. Su, H.-H. Huang, and C.-S. Lee, “Behaveior model simulations of micromouse and its application in intelligent mobile robot education,” *Proceedings of 2013 CACS International Automatic Control Conference*, Dec. 2-4, Taichung, Taiwan.

[17] H.-H. Huang, J.-H. Su, and C.-S. Lee, “A Contest-Oriented Project for Learning Intelligent Mobile Robots,” *IEEE Transactions on Education*, Vol. 56, No. 1, pp. 88-97, 2013.

[18] Juing-Huei Su, Chyi-Shyong Lee, Hsin-Hsiung Huang, Shih-Wei Chao, Sheng-Hong Lin, and Yu-Cheng Wu, “A Versatile Kit for Teaching Intelligent Mobile Robots,” *Next Wave in Robotics, Communications in Computer and Information Science*, Vol. 212, pp. 42-49, Springerlink, 2011.