Hands-on intelligent mobile robot laboratory with support from the industry

Hsin-Hsiung Huang, Juing-Huei Su, Chyi-Shyong Lee, Jheng-Yu Huang, and Sheng-Hsiung Chuang
Deptartment of Electronic Engineering
Lunghwa University of Science and Technology,
No. 300, Sec. 1, Wanshou Rd., Touyuan county, Taiwan
suhu@mail.lhu.edu.tw

Abstract—**The widespread use of robots in many areas makes the fundamental understanding of them a necessity for many electronic system design engineers. Therefore, to effectively speed up the learning process, the applications of learning-by-doing hands-on laboratory to help students get acquainted with the design and implementation of robots is inevitable. Lunghwa University has teamed up with local microcontroller manufacturers to redesign course contents, to host free workshops supported by the Ministry of Education, and to hold national contests for intelligent mobile robots. The devised low cost educational robot kits and multimedia lecture notes not only reinforce the hands-on laboratory exercises, but also help motivate students to learn actively the intelligent mobile robots.**

Keywords- mobile robots, hands-on laboratory

I. Introduction

Robots are attracting more and more people's attention recently, especially when Sony and iRobot announced the biped humanoid QRIO robot and Roomba vacuum cleaning robot, respectively. Robots are mechatronic engineering products, capable of acting autonomously in various physical environments. The widespread use of robots in many areas makes the fundamental understanding of them a necessity for many electronic system design engineers. Unfortunately, learning the design philosophy of robots is interesting but difficult, because it includes many areas of knowledge, e.g., the mechanics and electronics, linear and nonlinear vehicle control theory, and firmware programming of microcontrollers, etc. Teaching the autonomous mobile robot design course is a challenging undertaking because one can not assume that all students enrolled in the class have solid prerequisite knowledge in so many areas. It is noted by Heer *et al* [1] that introducing a platform for learning™ robotics courses enhances students' sense of community, innovation capabilities, and troubleshooting skills. The students could experience first-hand the fun associated with robots while gaining a sense of accomplishment. Therefore, to effectively inspire students' interest and motivate them to participate actively in the learning process, the application of hands-on laboratory which leads to an interesting robot to help students learn the design and implementation skills of robots is inevitable.

Heer *et al* [1] propose a low cost and easy to control Tekbots platform for hands-on freshman and even senior level robotic exercises. The Tekbots platform is simple and good for motivating the students, but it may not be used to experience more advanced firmware implementations, control, signal processing, and path planning algorithms. Kuc *et al* [2] used Javascript simulations to make students familiar with what robots can do to interact with environments. The students can develop their own logic circuit designs to program a robot for tasks of increasing complexity. Once the designs are verified in simulations, they can be downloaded to the robot and control the robot in an actual environment. Although students can check quickly whether their ideas work or not for tasks, this approach limits what students can learn in the hands-on exercises because the robot brain is a combinational logic circuit. To make the learning platform more fun and versatile, Lunghwa University of Science and Technology has teamed up with local microcontroller companies (see figure 1) to design low cost line following robots and micromouse, and the corresponding multimedia lecture notes. These companies support not only free samples of microcontrollers, integrated design environments (IDEs), but also free workshops and technical support in this project. By using a similar idea to that described by Hussmann *et al* [3] to inspire students' interest furthermore, the team also works together to hold contests for these mobile robots. The contests and free workshops for university and vocational high school students are also supported by the Ministry of Education. It should be noted that the contest[4] did attract many students and teachers. More than 200 teams compete with each other for the championships of fastest line following robots and micromouse this year.

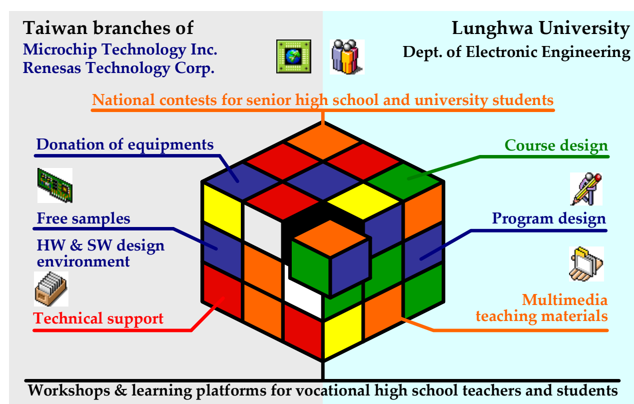


Fig. 1 The joint effort between Lunghwa University of Science and Technology and microcontroller manufacturers

II. Joint Efforts on Developing Learning Platforms

Learning by doing is a very effective idea for students to acquire knowledge, if they are well motivated. It is found that students will be willing to do tedious research work to solve practical problems when these problems are related to an interesting, and competitive contest [3]. Racing contests of line following robots and micromouse are such examples [4-5]. This is why Lunghwa University of Science and Technology wants to team up with local microcontroller companies to design low cost line following robots and micromice, and to hold jointly the corresponding contests. The followings are brief descriptions of the devised mobile robots.

A. Line following robots

Line-following robots are self operating robots that move along a line on the floor. Recently, the Fortune Institute of Technology had held 6 times of line following robot racing contests to encourage students from technological and vocational education system in Taiwan to show their creativity and techniques. It is a very successful contest not only because of the high prize, but also in attracting senior high school students to join the contest. The Taiwan micromouse and intelligent robot contest even include the line following robot racing contest as one of its major events this year [4].

The rules [4-5] allow contestants to memorize where straight lines and curve turns are along the track by providing prompts every time the curvature of the track changes. Figure 2 shows the 4th generation learning platform for line following robots of the joint effort. The idea [6] of using analog outputs of reflective optical sensors to estimate accurately the line position is applied in this prototype of learning platform. To calculate precisely the distance that the robot moves and lower down the cost, home-made encoders are installed onto two DC motors. The resolution is about 0.5mm/pulse. Two dsPIC microcontrollers are used in this configuration. One is for collecting the encoder pulse information for two dc servo motors, and the other is in charge of reading the reflective optical sensor outputs, calibrating these outputs, estimating the line position, and controlling the run speed and orientation of the robot. It is capable of running at speed more than 2m/sec. The robot costs about $ 150 USD. Two-third of the money is for the maxon DC motors due to its high performance and small size.

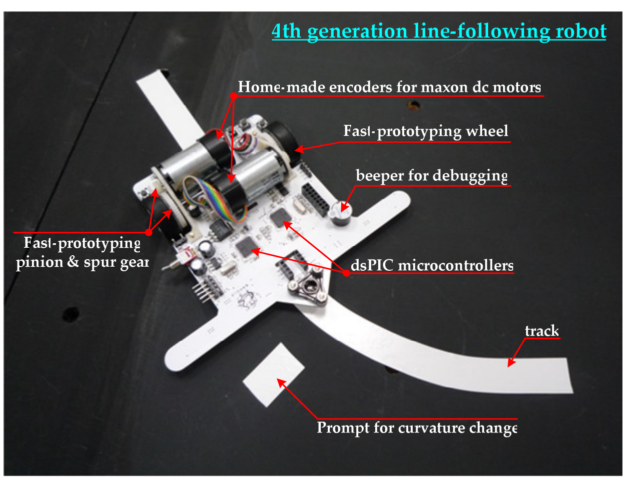


Fig. 2 A sample learning platform for line-following robots

B. Micromouse

Micromouse contests are for miniature autonomous robots to compete for the title of best speed and intelligence in solving a maze and finding the goal. Over the years, the micromouse contest became one of the few student competitions among engineering schools around the world. After all these years of development, the micromouse has made a great improvement in both maze-solving algorithms, and motion control implementation in the international competitions. This is partly due to the fast development of microcontrollers and sensors, which not only shrinks the hardware circuits, but also makes detect the environment conditions and motion control of the micromouse easier and execute the algorithms faster. These contests are still very popular to engineering students in UK [7], USA [8], Japan [5], Singapore [9], and Taiwan [4], because the micromouse project is helpful and fun for students to learn and integrate multidisciplinary knowledge. The micromouse is also a very good example for learning embedded control systems.

Figure 3 shows the 4th generation learning platform for micromouse contests of the joint effort. Local industries also contribute to make the maze court, the post, and the wall. The cost for the maze is about 30% of the price found on the Internet [10]. The functions of the two micro-controllers installed on the micromouse in figure 3 are basically the same as those for the line following robot in figure 2. Therefore, students familiar with controlling the line following robot in figure 2 can easily switch to this micromouse learning platform.

The differences between the micromouse and the line following robot that make the micromouse a more advanced project for intelligent mobile robots are threefold. They are 1) a maze-solving algorithm which should be integrated with the motion control algorithm, 2) various maze wall configurations should be taken into account in the motion control algorithm such that the micromouse won't hit the maze wall, 3) a gyro sensor should be used because the centrifugal force is large at high speed.

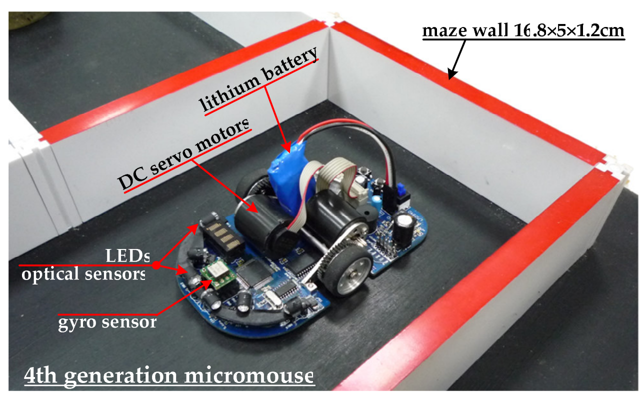


Fig. 3 A sample learning platform for micromouse contests

III. PROBLEMS STUDENTS CAN LEARN TO SOLVE IN THE LEARNING PLATFORMS

The followings are brief descriptions of some problems students may face in making their own line following robot and micromouse projects.

A. Mechanical Design of robots

Students that enrolled in department of electronic engineering in general do not have the opportunity to design their own mechanical systems. When students want to make their own line following robot or micromouse faster, they have to take into consideration the power, torque, and top speed ratings of their DC motors, the grip force between tyre and the ground, and the height of the center of gravity. These things are all related to the mobility of the robot. This motivates the students to learn computer aided software to help them design and verify their own mechanical structures before construction. Figure 4 shows such examples for mechanical designs of stepper motor and dc servo motor micromice. Although these are only simple structures and won't take students too much time to learn to design their own, they do bring a lot of fun to students in designing their own robots.

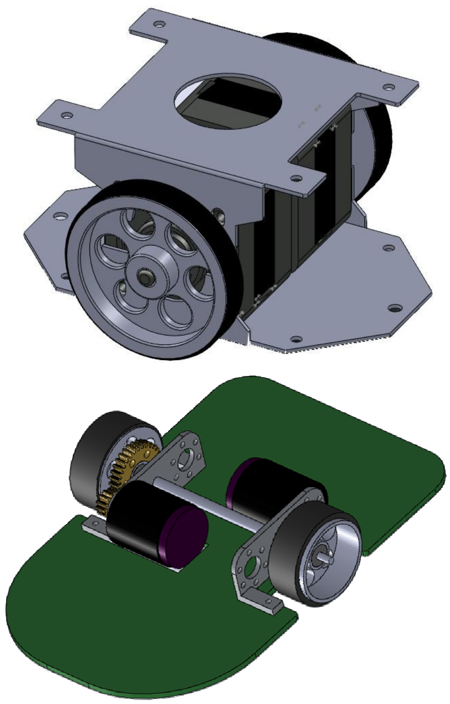
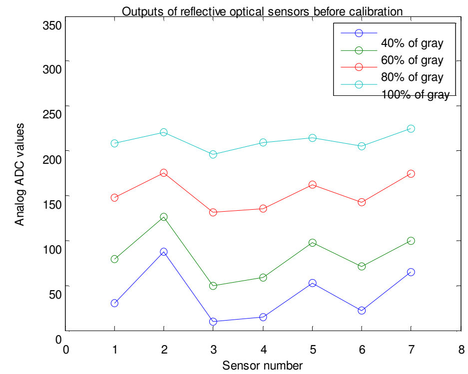


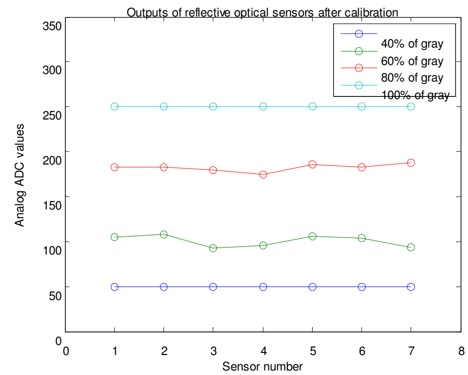
Fig. 4 Example student designs of mechanical structures for micromice

B. Calibration of Optical Sensor Outputs and line detection

To detect the line in the racing track, most people use two or more reflective optical sensors[11]. Unfortunately, this approach works fine only at low speed. Chan [6] uses analog output voltages from reflective optical sensors and digital PD control algorithms, such that the line following robot can run faster than 80cm/sec. Before students can use the analog sensor outputs to estimate line positions, each output should be calibrated to give almost the same level of voltage under the same working condition. This is due to the variations of optical sensor characteristics, even though their part numbers are the same. It can be seen in figure 5a that the output values of different reflective optical sensors vary a lot even if they are under the same working condition. Figure 5b shows the results after the software calibration procedure [12] is applied.



(a)



(b)

Fig. 5 The reflective optical sensor outputs for different gray scale (a) before and (b) after calibration

After the calibration procedure is finished, it is ready to calculate the line position based on those adjusted analog output values from the corresponding reflective optical sensor circuits. Students can use various interpolation techniques or the way of find the "center of mass" [14] to do it. It is important at this part for students to figure out a way to verify how accurate their estimation algorithm is.

Suppose that the coordinate of the 7 reflective optical sensors are x_0, x_1, x_2, x_3, x_4, x_5, x_6, respectively, and the corresponding analog output values are y_0, y_1, y_2, y_3, y_4, y_5, y_6, which is shown in figure 7. The estimated line position can then be calculated by the following weighted average formula:

$$x = \frac{\sum_{i=0}^{7} x_i y_i}{\sum_{i=0}^{7} y_i} = \frac{3(y_6 - y_0) + 2(y_5 - y_1) + (y_4 - y_2)}{\sum_{i=0}^{7} y_i}, \qquad (1)$$

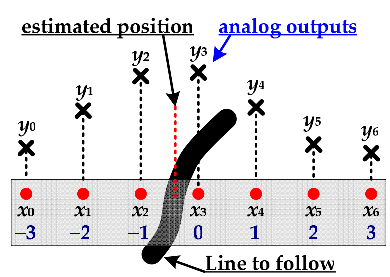


Figure 6 The line detection algorithm via weighted average

Figure 7 shows the experimental results by using the "center of mass" approach.

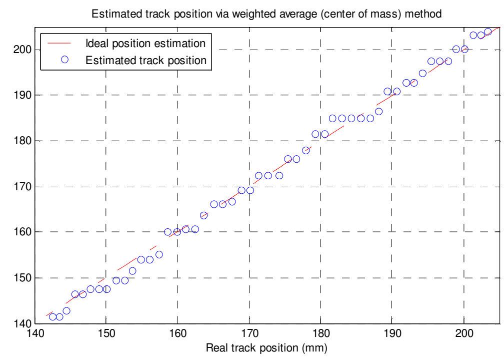
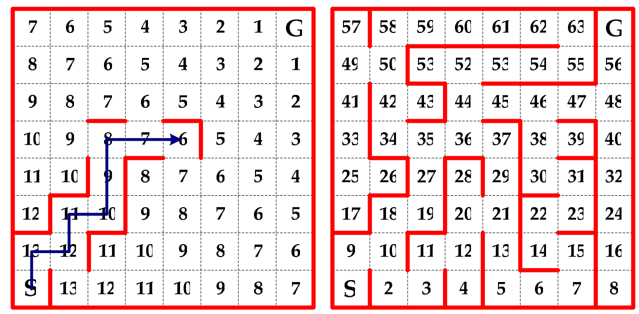


Figure 7 The experimental verification of the center of mass line detection algorithm

C. Maze solving algorithms

To solve a maze and plan a shortest path from the start point to the goal, the micromouse has to search and memorize the environment first, and then find the shortest path based on some optimization approaches. The simplest method available to a micromouse is some variation on the flood-fill or Bellman algorithm [15-16]. The idea is to start at the goal square and fill the maze with values which represent the distance from each square in the maze to the start or goal square. When the flooding reaches the goal, or start square, the algorithm can then be stopped. The micromouse could follow the values downhill or uphill to the goal square. Take the 8×8 maze shown in figure 8a as an example. The number at each square in the maze in figure 8b is the index for the square stored in the array. At first, the micromouse does not know any information about the maze configuration, except the start square. The first iteration of the flood algorithm then gives the result shown in figure 8a, if the distance value is to the goal square. The micromouse then tries to follow the values downhill to the goal square. Since the distance value to the goal square of the start square is '14', the micromouse will try to move itself to a square whose distance value is '13'. Because the square no. 2 is blocked by a maze wall, the micromouse goes from the start square to the square no. 9. The process goes on and on until the micromouse reaches the square no. 37 whose distance value is '6' as shown in figure 1a, because the micromouse can not find a way to go to squares whose distance value is '5'. Under this condition, the flood algorithm should be executed once again. The distance values in that array would be updated according to the updated maze wall information shown in figure 8a. By using the procedure described above, the goal square would be found at last.

A more advanced flood algorithm should take not only the distance between the start square and the goal square into account, but also the movement of 'going straight' or 'turning'. In other words, the time for the micromouse to move from one square to another should also be considered in an advanced flood algorithm. It is because the weight transfer effect [18] would make the micromouse skid if it makes turns too fast. Therefore the speed of the micromouse should be slower in making turns than in going straight. The above idea of using different weights on the distance recorded in each square to the goal square may also reduce possible paths after the flood algorithm is executed.



(a) (b)

Figure 8 (a) Solving the maze via the flood algorithm if the distance recorded in each cell is the distance to the goal square, (b) square indices of the maze.

When the goal square and the shortest paths from the start square to the goal square are found, the shortest paths can be furthermore optimized such that the micromouse can run in diagonal instead of consecutive 90 degree turns to reduce even more the run time from the start square to the goal square. Figure 9 shows such 3 possible paths with diagonal routes and 45 degree turns.

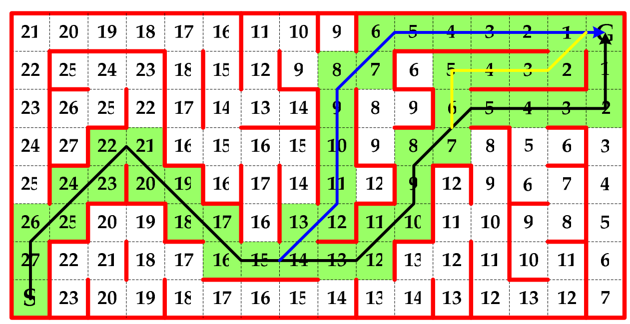


Figure 9 Using diagonal runs to optimize given shortest paths

To speed up the learning process, the maze solving algorithms described above are implemented as a simulation program which is shouwn in figure 10. The input information for maze solving algorithms is the maze wall configurations. The algorithms will then generated distance values from any square in the maze to the goal square. These values will then be used to find out the optimal route for a micromouse to go from the start square to the goal square. The user interface shown in figure 10 is convenient for students not only to define various mazes, but also to debug their codes for the algorithms.

C. Motion control algorithms

The results of the maze solving algorithm can then be used in the motion control algorithm to drive the micromouse as fast as possible from the start point to the goal. Although the optimal path with diagonal segments can be drawn by using lines (see figure 9), the trajectory for the micromouse should be smooth enough such that it won't skid due to centrifugal force when making turns.

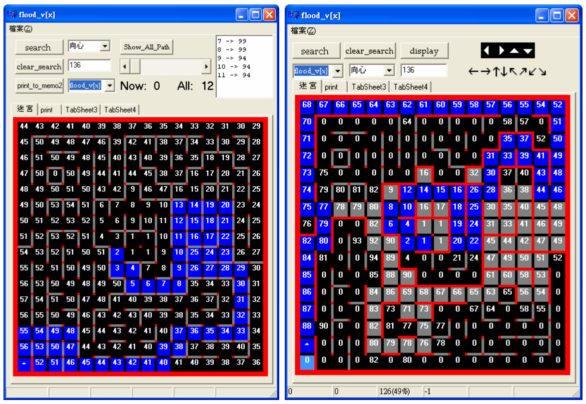


Figure 10 User interface of the maze-solving program

Therefore, smooth trajectory patterns for the micromouse when making 45, 90, 135 degree turns, and making U and V turns shown in figure 11 should be stored in the firmware of the micromouse. The stored information for these smooth trajectories includes the diameter of wheels, start and end positions, and the speed profiles for both wheels. The diameter of the wheel is necessary because the firmware can only track the steps of motors, and the distance traveled is also related to the diameter of wheels.

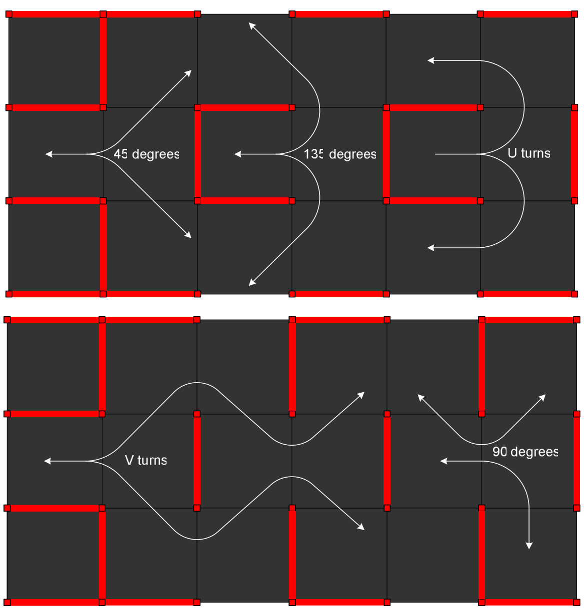


Figure 11 Smooth trajectory patterns for micromouse to make 45, 90, 135 degree, U and V turns.

The approximate mathematical equations proposed in [17] can be applied to find out the position of the micromouse in the maze:

$$v_L(k) = \pi D \times \frac{p_L(k)}{T_s \times p_C},\ v_R(k) = \pi D \times \frac{p_R(k)}{T_s \times p_C}, \quad (2.a)$$

$$v(k) = \frac{v_L(k) + v_R(k)}{2},\ \omega(k) = \frac{v_R(k) - v_L(k)}{L}, \quad (2.b)$$

$$x(k+1) = x(k) + T_s v \cos(\theta(k)), \quad (2.c)$$

$$y(k+1) = y(k) + T_s v \sin(\theta(k)), \quad (2.d)$$

$$\theta(k+1) = \theta(k) + T_s \omega(k+1), \quad (2.e)$$

where D, $p_L(k)$, $p_R(k)$, p_C, $v(k)$, $v_L(k)$, $v_R(k)$, $\omega(k)$, L, $x(k)$, $y(k)$, $\theta(k)$, and T_s stand for the diameter of the wheel, the pulse counts of left wheel at the kth sampling interval, the pulse counts of right wheel at the kth sampling interval, the pulse counts of the stepper motor to run a round, the velocity of the micromouse at the kth sampling interval, the velocity of left wheel, the velocity of the right wheel, the angular velocity of the micromouse, the distance between the wheels, the x, y coordinate of the midpoint between the wheels, the orientation angle of the micromouse, and the sampling interval for the digital control algorithm, respectively. The formula in (2) is accurate enough if the sampling interval is sufficiently small.

Although stepper motors are used in the micromouse proposed in this paper, the same pulse commands for the two stepper motors can not guarantee that the micromouse will go in a straight line. This is because the load conditions and the friction forces for the two wheels are eventually not the same. Moreover, the 45, 90, 135, 180 degree turns based on the pulse commands, circumferences of the wheels, and the distance between the wheels also need to be corrected due to measurement errors. To make the micromouse run smoothly in a given maze, there are three types of errors which need to be corrected based on the optical sensors installed on the micromouse. They are 1) longitudinal error, 2) lateral error, and 3) alignment error, which are shown in figure 12.

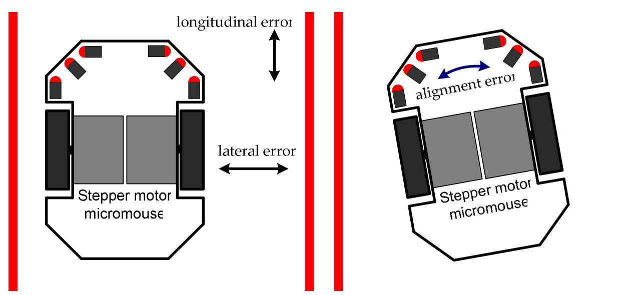


Figure 12 Error sources for a micromouse in a given maze

The alignment and lateral errors of the micromouse can be corrected by using either the pulse commands or the infrared sensor readings. The longitudinal error can be corrected if the pole position of the maze can be detected when the micromouse runs in a given maze. This is shown in figure 13. Because the initial position and the orientation angle θ of the micromouse are all set to 0, these calculation errors should also be corrected based on the infrared sensor readings. Diagonal straight line motion can be corrected by using the readings of forward-looking optical sensors, such that the micromouse would not hit the posts or corners alongside the path.

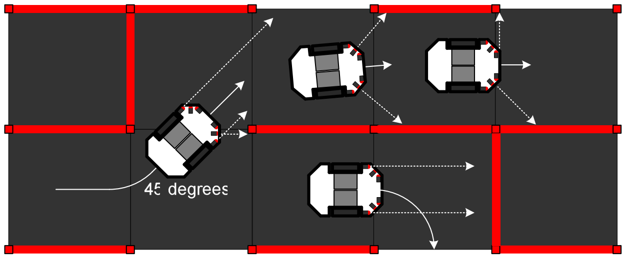
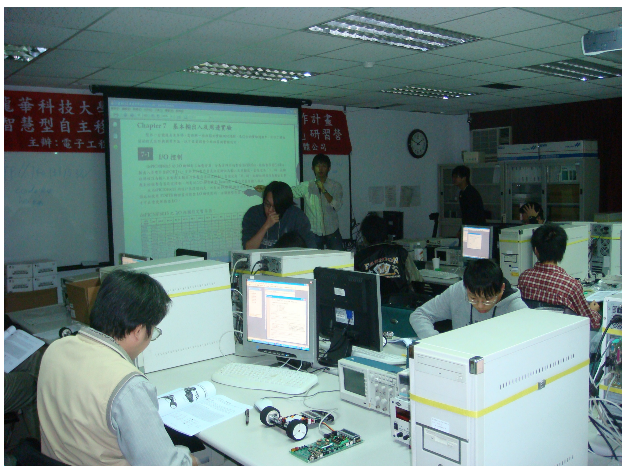


Figure 13 The strategies that a micromouse can use in correcting the longitudinal, lateral, and alignment errors

D. Workshops for vocational high school teachers and students

The joint effort described in this paper is also used to encourage vocational high school teachers and students to learn implementation skills of intelligent mobile robots. Figure 14 shows the free workshops financially supported by Ministry of Education. Lecturers include students, teachers of Lunghwa University of Science and Technology, and technicians from local microcontroller manufacturers. Although most of the students are teaching assistants, there are indeed some students who are proficient enough acting as lecturers. Moreover, every one of them who joins the workshop can bring back not only the intelligent mobile robots, but also the IDEs.


(a)


(b)

Figure 14 Free workshops of (a) line-following robots, and (b) micromouse for vocational high school students

IV. CONTESTS FOR INTELLIGENT MOBILE ROBOTS

To see how well students integrate all the skills learned in the hands on laboratory in implementing line-following robots and micromice, contests (see figure 15) were held after they finished their intelligent mobile robots. Students are also encouraged to join national contests [4]. It is interesting to note that the racing contest did motivate the students to strive to learn the knowledge and skills necessary to make intelligent mobile robots. For example, one group of students in line following robots tried changing the distance between consecutive optical sensors and obtained better results in predicting the line position. They even made a new line-following robot (shown in figure 16) which could run at a maximum speed of 1.2m/s. This is a competitive design to those devised by Cook [13].


(a)


(b)

Figure 15 Contests for (a) line-following robots, and (b) micromouse.

V. QUESTIONNAIRE FEEDBACK

A survey was conducted in a hands-on laboratory for line following robots. It can be seen from Table 1 that the feedback on the questionnaires from the students was quite positive. The majority of the students (78%) agreed that they were motivated to learn those skills and theories, and were satisfied (90%) with the organization of the laboratory. The feedback also shows that the project-based laboratory was more appealing than an earlier conventional laboratory. However, there are three students who thought that the cost is still too high for them, which is partly due to the fact that most of the department's students are economically disadvantaged.

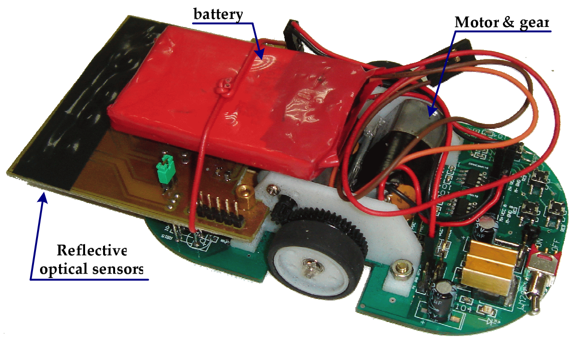


Figure 16 The new line-following robot devised by students

Table 1. The average scores of the survey for the hands-on laboratory

Questions	Average
The hands-on laboratory can effectively help me learn hardware circuit and programming implementation skills.	**4.06 (1-5)**
The contests held after the mobile robots are finished interest me a lot and encourage me to learn more about the necessary skills and theories.	**3.91 (1-5)**
It is more interesting to me to make the robot from scratch in both hardware and software, because each step is explained in detail.	**4.15 (1-5)**
I am willing to pay the money to own the mobile robot.	**3.52 (1-5)**
I am satisfied with the project-based hands-on laboratory.	**4.50 (1-5)**

VI. CONCLUSIONS

The joint effort of Lunghwa University and local microcontroller manufacturers to redesign course contents and learning platforms, to host free workshops for vocational high school teachers and students, and to hold national contests for intelligent mobile robots are described in this paper. The effort was also appreciated and financially supported by the Ministry of Education. The local microcontroller manufacturers support not only free chips and IDEs, but also technicians to help teach in workshops. The effort is seen to effectively motivate students in classes and workshops to learn actively the intelligent mobile robots.

ACKNOWLEDGMENT(S)

The authors would like to thank National Science Council for its financial support under grant NSC 98-2511-S-262 -002.

REFERENCES

[1] D. Heer, R. L. Traylor, T. Thompson, and T. S. Fiez, "Enhancing the Freshman and Sophomore ECE Student Experience Using a Platform for Learning™," *IEEE Transactions on Education*, Vol. 46, No. 4, pp. 434-443, Nov. 2003.

[2] R. Kuc, E. W. Jackson, and A. Kuc, "Teaching Introductory Autonomous Robotics With JavaScript Simulations and Actual Robots," *IEEE Transactions on Education*, Vol. 47, No. 7, pp. 74-82, Feb. 2004.

[3] S. Hussmann and D. Jensen, "Crazy car race contest: multicourse design curricula in embedded system design," *IEEE Transactions on Education*, Vol. 50, No. 1, pp.61-67, Feb. 2007.

[4] Lunghwa University of Science and Technology, The 5th Taiwan micromouse and intelligent robot contest, http://robot2009.lhu.edu.tw/, accessed on 8/15/2009.

[5] New technology foundation, All Japan micromouse and robotrace contests, accessed on 8/15/2009, http://www.robomedia.org/directory/jp/game/index.html.

[6] Chan, "Desktop line following robot," http://elm-chan.org/works/ltc/report.html, accessed on 5/8/2009.

[7] UK micromouse competition 2008, http://www.tic.ac.uk/micromouse, accessed on 5/29/2009.

[8] The 22nd annual micromouse contest in the *IEEE Appllied Power Electronics Conference and Exposition 2008*, http://www.apec-conf.org/2008/index.php?option=com_content&task=view&id=7&Itemid=168, accessed on Aug. 18, 2009.

[9] Singapore Robotic Games 2010, http://guppy.mpe.nus.edu.sg/srg/, accessed on Aug. 18, 2009.

[10] Robot Store (HK), Micromouse Maze Parts, http://www.robotstorehk.com/micromouse/RS-MAZE.html, accessed on Aug. 18, 2009.

[11] J. Vroman, "JavaBot1...A line following robot,", http://www.james.vroman.com/javbot1a.htm, accessed on Aug. 15, 2009.

[12] C.-S. Lee, J.-H. Su, K.-E. Lin, J.-H. Chang, M.-H. Chiu, and G.-H. Lin, "A hands-on laboratory for autonomous mobile robot design courses," the *17th int. federation of automatic control world congress*, pp. 9473-9478, July 6-11, Seoul, Korea, 2008.

[13] David Cook, ROBOT ROOM: *Jet- the ultra-fast line-following robot*, http://www.robotroom.com/Jet.html, Accessed on August 19, 2009.

[14] Wikipedia, (2009) Center of mass – alternative derivation, http://en.wikipedia.org/wiki/Center_of_mass, accessed on June 14, 2009.

[15] Peter Harrison, Micromouse Resources, http://micromouse.cannock.ac.uk, accessed on 5/29/2009.

[16] Society of Robots, "Micromouse – Maze Solver – Theory," accessed April 15, 2009, http://www.societyofrobots.com/member_tutorials/node/94.

[17] T.-C. Lee, K.-T. Song, C.-H. Lee, and C.-C. Teng, "Tracking control of unicycle-modeled mobile robots using a saturation feedback controller," *IEEE Trans. on Contr. System and Tech.*, vol. 9, pp. 305–318, March, 2001.

[18] TurnFast.com, "Weight Transfer," accessed on 4/15/2009, available at http://www.turnfast.com/tech_handling/handling_weightxfr.