A Project-Based Laboratory for Learning Embedded System Design With Industry Support

Chyi-Shyong Lee, Juing-Huei Su, *Senior Member, IEEE*, Kuo-En Lin, Jia-Hao Chang, and Gu-Hong Lin

***Abstract*—A project-based laboratory for learning embedded system design with support from industry is presented in this paper. The aim of this laboratory is to motivate students to learn the building blocks of embedded systems and practical control algorithms by constructing a line-following robot using the quadratic interpolation technique to predict the line position. For those students who have acquired basic microcontroller hardware and software programming skills from previous courses, the hands-on exercises in the laboratory include several specific hardware circuits and software algorithms for the final project of constructing the line-following robot. The students are allowed to discuss the hardware and software problems with each other while solving each exercise, although they have to answer the teacher's questions individually to earn the score. To enhance the learning outcomes, a racing contest for the students' line-following robots is also organized to see how well the techniques learned in the laboratory are applied in the final project. The support from the local branch of Microchip Inc. allows students to obtain C-compilers and microcontrollers at no cost. The feedback from students shows that the final project of constructing line-following robots and the racing contest motivates the students to learn actively all the skills included in the laboratory for embedded system design.**

***Index Terms*—Embedded system, line-following robot, project-based.**

I. Introduction

EMBEDDED systems are widely used nowadays in such areas as sensing, communications, command, and control thanks to the fast development of microelectromechanical (MEMS) sensors, integrated circuits for wireless communication technologies and microcontrollers, and so on [1]–[3]. These systems measure the temperature, humidity, position, or velocity of mechanical systems and detect human motions or responses. They also control motors and other devices and deliver information for manufacturing. *The New York Times* estimated several years ago that the average American came into contact with 100 microcontrollers per day [1]. The opportunities for embedded system design seem to be endless and are limited only by the availability of creative ideas and novel technologies. Therefore, it is essential for students and engineers to learn from a system perspective [1] how to design embedded systems. Unfortunately, while learning the design philosophy of embedded systems is interesting, it is also difficult because it includes many areas of knowledge [4], e.g., microcontroller structures and programming [1]–[3], [5], [6], interfacing technologies [1], [4], automatic control theory [2], [5], and sensor technologies [2], [3], [5], etc. Teaching embedded system design is a challenging undertaking because a teacher cannot assume that all students enrolled in a class have solid prerequisite knowledge across all these areas. To speed up the learning process and motivate students to learn actively, the project-based learning approach [3], [5], [6] is applied in this embedded system design laboratory. Although Hussmann and Jensen [5] proposed successful multicourse design curricula in embedded system design based on the "Crazy Car Race Contest," the necessary hardware and software facilities for the courses are quite expensive. An integrated development environment for 32-bit ARM-7 microcontrollers and Xilinx Complex Programmable Logic Devices costs about $700 USD in the Department of Electronic Engineering, Lunghwa University of Science and Technology, Taiwan. Given the availability of this environment, an affordable ($\sim$ $65 USD) and versatile line-following robot kit was also devised for the exercises and as a final project in this laboratory.

The low-cost line-following robot kit proposed in this paper serves as a good example on which students can learn embedded system design skills because it covers not only common embedded system peripherals but also energy control and quadratic line detection with sensor-calibration algorithms and real-time control firmware implementation. The quadratic line-detection with sensor-calibration algorithm plays an important role in rapidly driving the line-following robot with digital control algorithms. This approach differs from that proposed in [7] in the ways used to calibrate the sensor outputs, to predict the line position according to the calibrated sensor outputs, and to control the energy consumed by the hardware sensor circuits. Although usable parameters for the algorithm are not difficult to obtain, deliberate study of the hardware sensor circuits and the calibration procedure pays off in performance improvements. The process of fine-tuning the parameters of the algorithm can give students the idea that hardware circuits and software algorithms are both important for a successful embedded system design.

The competition between student groups in the racing contest can also motivate the students to explore in depth the skills acquired in this laboratory as well as give them lots of fun. The students can participate not only in the local racing contest held in school but also in annual national [8] and international [9] racing contests. This opportunity engages the students even more, both because the rules give hints every time the curvature of the track

Manuscript received April 24, 2008; revised October 22, 2008. First published June 30, 2009; current version published May 05, 2010. This work was supported by the National Science Council, Taiwan, under Grant NSC 96-2516-S-262-002-MY2.

The authors are with the Department of Electronic Engineering, Lunghwa University of Science and Technology, Taoyuan 33306, Taiwan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2008.2010990

changes, thus creating lots of room for further improvements, and also because of the significant prize associated with the contest. Students' motivation for cooperative design and active lifelong learning about embedded systems will also be improved compared with previous works proposed in [2], [3], and [5]. Details of the project-based hands-on laboratory on embedded system design are presented in this paper such that university departments, even those with stringent budget constraints, can easily redesign the kit and courseware for their students.

II. The Project-Based Laboratory for Embedded System Design

The project-based laboratory for embedded system design evolved from an optional senior undergraduate course in systems design with digital signal processors in the Department of Electronic Engineering, Lunghwa University of Science and Technology. The earlier course ran conventional laboratories for learning and understanding course materials. The prerequisite subjects for this laboratory are computer programming and single-chip microcontrollers, which are compulsory subjects for the first- and third-year students, respectively. The laboratory is also supported by the local branch of Microchip Inc., and students can obtain C-compilers and microcontrollers used in this laboratory at no cost.

The objective of this project-based hands-on laboratory is to introduce components and procedures for designing embedded systems, which are then integrated by students to implement their final projects of line-following robots. The topics covered in the hands-on project-based laboratory using the proposed low-cost and versatile line-following robot kit include the following:

- the integrated development environment for microcontrollers;
- interrupt-driven programming for task scheduling;
- the calibration procedure for sensor outputs;
- the quadratic line-detection algorithm;
- pulse-width-modulation (PWM) speed control strategy for dc motors;
- the tracking control algorithm, behavior model simulation, and implementation.

Sample firmware codes are given to students for these topics so that they will be able to write their own codes faster once the ideas and practical implementation skills are introduced. These topics for constructing the embedded control system for the line-following robot project are briefly described as follows.

A. The Integrated Development Environment for Microcontrollers

Since the students who join the project-based hands-on laboratory are expected to have preliminary knowledge about microcontrollers and programming skills, this laboratory section is mainly used to make the students familiar with the functions that the dsPIC microcontrollers provide and the necessary documentation [10]. Students are asked to display characters on a liquid crystal display (LCD), which is shown in Fig. 1. This task serves to remind the students how the high level C language, the complier, linker, debugger, and the basic input and output subroutines for the microcontroller are used.

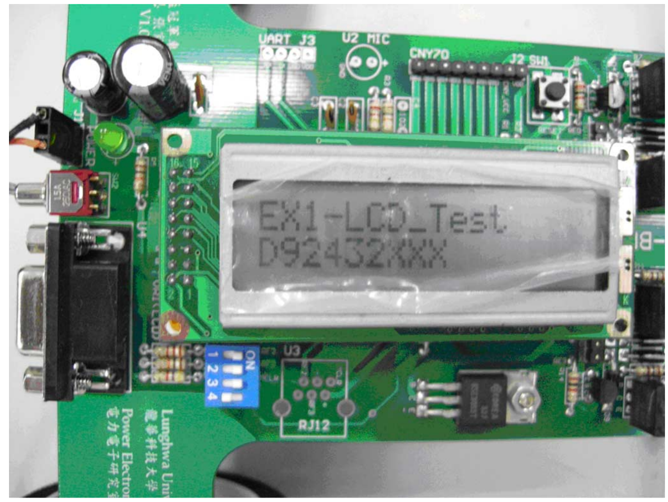


Fig. 1. The first section of the hands-on laboratory: Displaying characters on an LCD.

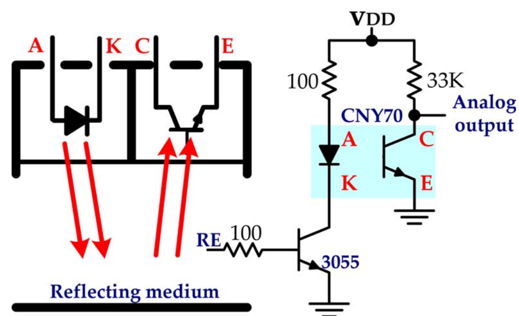


Fig. 2. The reflective optical sensor circuit in the line-following robot kit.

B. Interrupt-Driven Software Programming

Most embedded system applications need to react to the inputs or environment changes in real time, which means that the accuracy of computations is as important as their timeliness [1], [2]. Moreover, digital control algorithms need a fixed sampling time interval for measuring inputs and delivering output commands [11]. Therefore, the idea of applying interrupts for task scheduling is introduced.

The analog-to-digital conversion (ADC) function is used as an example in this section of the hands-on laboratory. The students should demonstrate on the LCD how large the analog input voltage is and refresh the value at fixed periods. The following topics are covered in this section:

- how the timer modules can serve as time counters and free-running interval timers or counters;
- how to define a scheduled period for real-time applications;
- how the corresponding interrupt service routines are programmed when the timers generate interrupts on period match;
- how the priority value for every interrupt is set and used to schedule different interrupts;
- how to configure the corresponding registers and read the ADC ports consecutively or simultaneously.

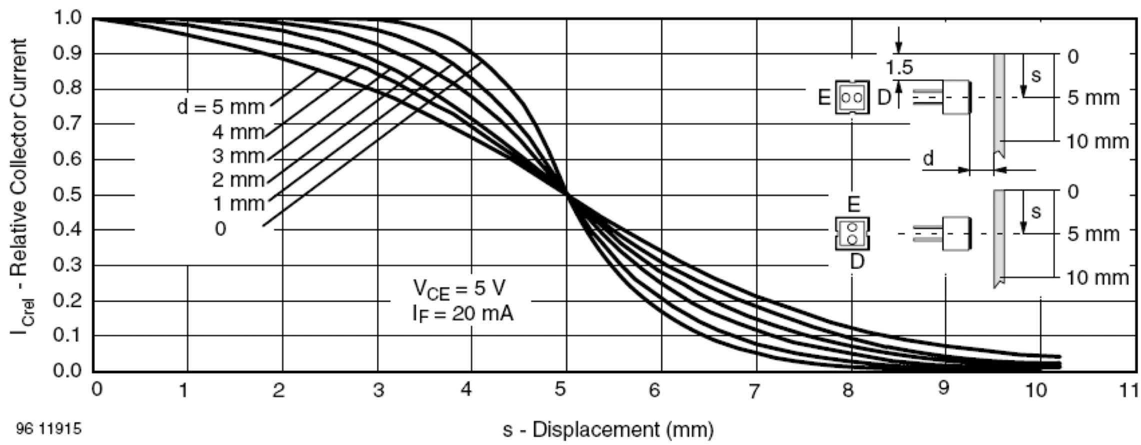


Fig. 3. The relationship between the reflective collector current and the displacement of the sensor from the line.

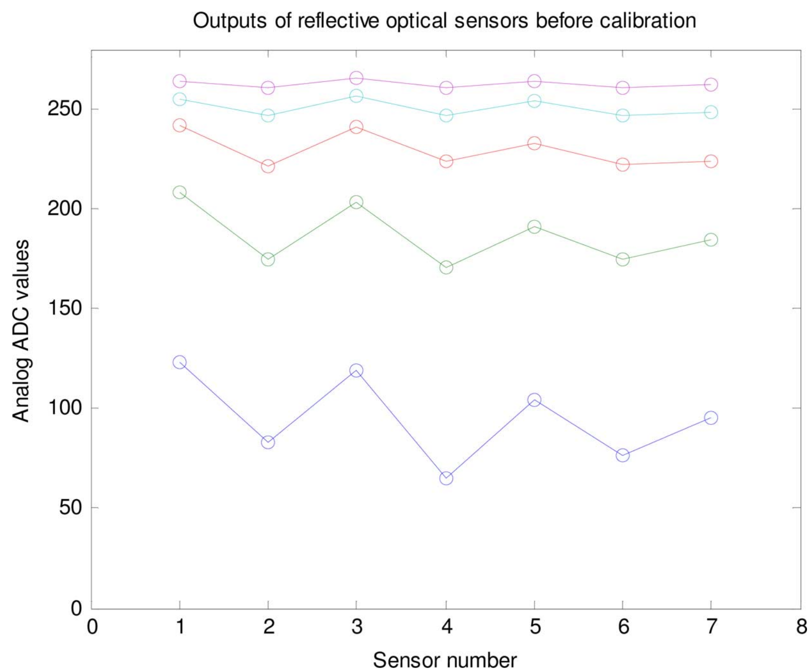


Fig. 4. The reflective optical sensor outputs for different gray scale (20%, 40%, 60%, 80%, 100%) before calibration.

C. The Calibration Procedure for Sensor Outputs

The use of reflective optical sensors is introduced in this section. Students should learn that optical sensors, even if they have the same part number, may not possess the same characteristics. Therefore, the analog outputs of each reflective optical sensor circuit shown in Fig. 2 should be calibrated such that these output values can later be used to determine the correct line position. As can be seen in Fig. 3, [12], the reflective collector current of the circuit in Fig. 2 can provide the information of how far the reflective optical sensor is from the line. Therefore, the output voltage would gradually change as the reflective optical sensor moves away from the line. This effect is the main idea used in [7] for line-following robots.

The calibration procedure starts by using a reference gray scale pattern to collect the analog outputs of different reflective optical sensor circuits. The values in Fig. 4 show that seven reflective optical sensors actually do not give identical outputs even under the same test conditions. The output values of each reflective optical sensor with respect to 20% and 80% of gray are then mapped to the same maximum and minimum values by using the following formula:

$$y_{jo} = y_{\min} + \frac{y_{\max} - y_{\min}}{x_{\max,i} - x_{\min,i}}(x_{ji} - x_{\min,i}) \qquad (1)$$

where $x_{\max,i}, x_{\min,i}$ are the maximum and minimum values of ith reflective optical sensor outputs, $y_{\max}, y_{\min}$ are the predefined maximum and minimum values for all reflective optical sensor outputs, x_{ji} is the jth output value for the ith reflective optical sensor, and y_{jo} is the calibrated output value for x_{ji}. It can be seen in Fig. 5 that the output values of different reflective optical sensors are quite similar to one another after the calibration procedure is done by using (1).

To avoid influences from the other light sources on the analog outputs of the reflective optical sensors in Fig. 2 and to use energy more efficiently, the current flow through the reflective op-

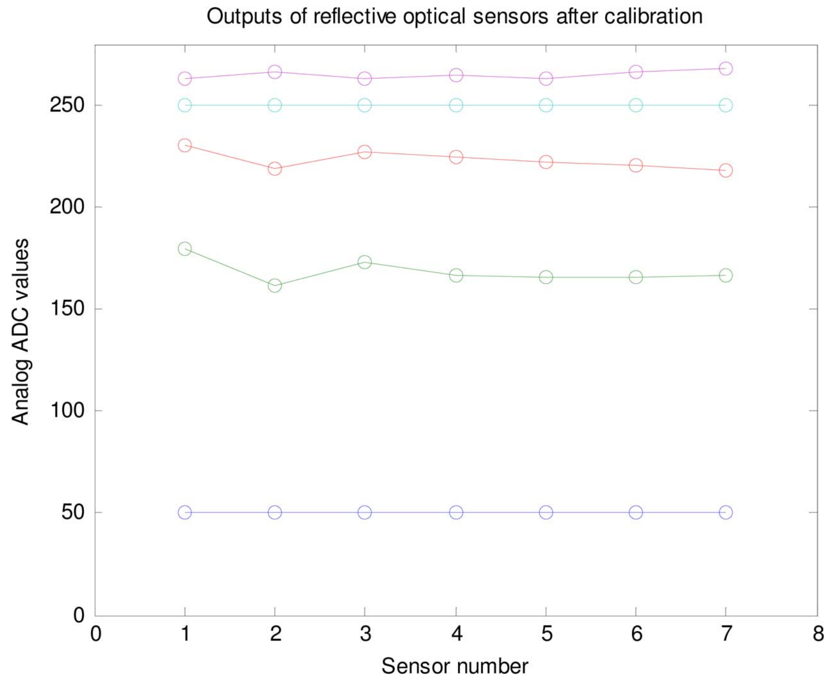


Fig. 5. The reflective optical sensor outputs for different gray scale (20%, 40%, 60%, 80%, 100%) after calibration.

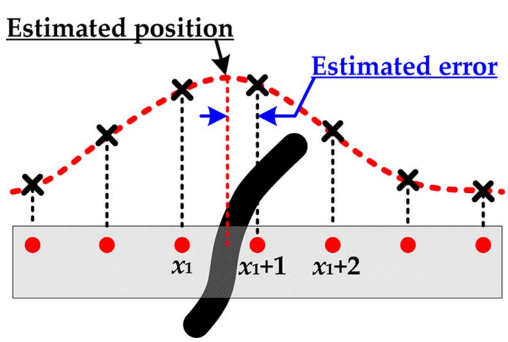


Fig. 6. The output values of the seven reflective optical sensors and the line-estimation algorithm.

tical sensors is controlled via a transistor switch (3055) shown in Fig. 2. This approach is taken because readings from the ADC ports of the microcontroller before and after the transistor switch is turned on could reveal the influences coming from the other light sources.

D. The Quadratic Line-Detection Algorithm

Because the final project in this hands-on laboratory for the students is to construct a line-following robot and compete with the other groups in a racing contest, a better way of detecting the line position, compared to other simple line-following robots [13], by using a quadratic interpolation technique is introduced. Assume that seven reflective optical sensors are used, and the coordinate of the leftmost sensor is 0. To find out the correct position of the black line, the students have to locate three consecutive sensors with higher output readings than the other four sensors, as shown in Fig. 6.

Suppose that the coordinates of these three sensors are x_1, x_1+1, and x_1+2, and the true shape of the sensor output values in the range of $[x_1, x_1+2]$ can be approximated by a quadratic curve. The following relationships can then be found between the coordinate of the sensors (x_1, x_1+1, x_1+2) and the output values (y_1, y_2, y_3):

$$y_1 = ax_1^2 + bx_1 + c$$

$$y_2 = a(x_1+1)^2 + b(x_1+1) + c \quad \text{(2a)}$$

$$y_3 = a(x_1+2)^2 + b(x_1+2) + c. \quad \text{(2b)}$$

The coordinate value at which the output value of the quadratic curve is the maximum is considered as the true position of the line. By using the basics of calculus, the coordinate value for the line position is known to be

$$x = -\frac{b}{2a}. \quad (3)$$

From (2) and (3), the true line position can be estimated by using the following formula:

$$a = \frac{y_1 + y_3 - 2y_2}{2}, \quad b = y_2 - y_1 - 2ax_1 - a. \quad (4)$$

The students should show the comparisons of the true line positions and the estimations by using (2)–(4) with and without the proposed calibration procedure on the reflective optical sensor outputs. It can be clearly seen in Fig. 7 that the calibration process plays a significant role in the line-detection algorithm, and the estimation is quite satisfactory. This calculation is also a practical implementation issue with respect to variations of sensor characteristics that students can learn in this laboratory.

E. The Pulse-Width-Modulation (PWM) Speed Control Strategy for dc Motors

The PWM control strategy is a well-known technique in which microcontrollers control the dc/ac motors via digital signals [14] and an H-bridge 4 quadrant inverter (shown in Fig. 8). If a bipolar scheme is used, the dc motor will stop and

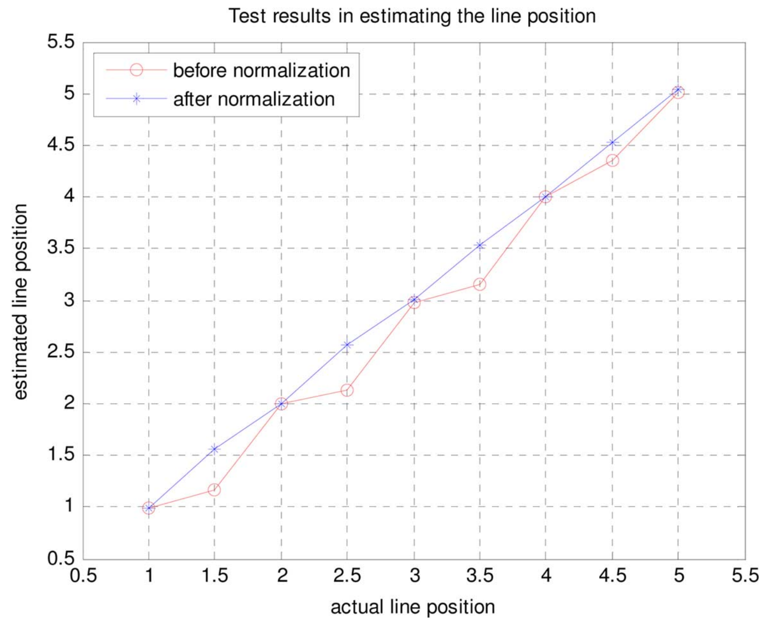
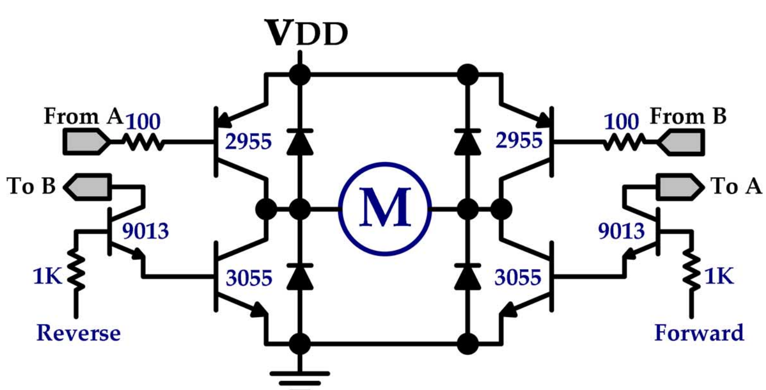


Fig. 7. Comparisons of the true line position and estimations with and without the sensor calibration process.



Forward	Reverse	Motor Status
1	0	Forward
0	1	Reverse
0	0	Uncontrolled
1	1	Not allowed

Fig. 8. The low-cost H-bridge converter for driving dc motors in this hands-on laboratory.

be held still when the duty cycle of the PWM is 50%. As the duty cycle increases beyond 50%, the speed of the dc motor increases in one direction until it reaches the maximum speed at a duty cycle of 100%. Conversely, the dc motor will reach the maximum speed in the other direction when the duty cycle decreases to 0%. Therefore, the speed and rotation of a dc motor can be adjusted by PWM signals from the microcontroller. Because there is a turn-on and turn-off delay time for the power transistors in a leg of the H-bridge inverter shown in Fig. 8, a “deadtime” should be inserted in the PWM control signals [15], [16] to prevent the so-called “shoot-through” between the dc source and ground. This implementation issue will also be introduced to the students in this section.

F. The Tracking Control Algorithm, Behavior Model Simulation, and Implementation

Control algorithms are often necessary in embedded systems. The popular proportional-integral-derivative (PID) controller is introduced in this section. To teach students how to apply the PID control algorithm to drive the robot along a line, SIMULINK behavior model simulations are used to familiarize students with the car dynamics and the effect of the PID controller on adjusting the car position. The model shown in Fig. 9 simulates the dynamics of the line-following robot whose inputs are the voltage commands for the dc gear motors, and outputs are the plenary coordinates and orientation of the line-following robot. Low-pass filters are used to simulate the responses of the dc gear motor speeds with respect to input voltages. The error between the center of the reflective optical sensors and the line to be followed is then processed by the PD controller in Fig. 10 to generate velocity commands for the right and left wheels. The D term will be required in most cases to stabilize tracking motion. Although integral control is not used, the steady-state error is small enough that it does not cause too much trouble.

The simulation results in Fig. 11 show the performance comparisons of the P and PD controller. It can be seen that if the gain values of the PD controller are carefully adjusted, the PD controller is better than the P controller, and the line-following robot can still follow a reference trajectory with an abrupt change of its path.

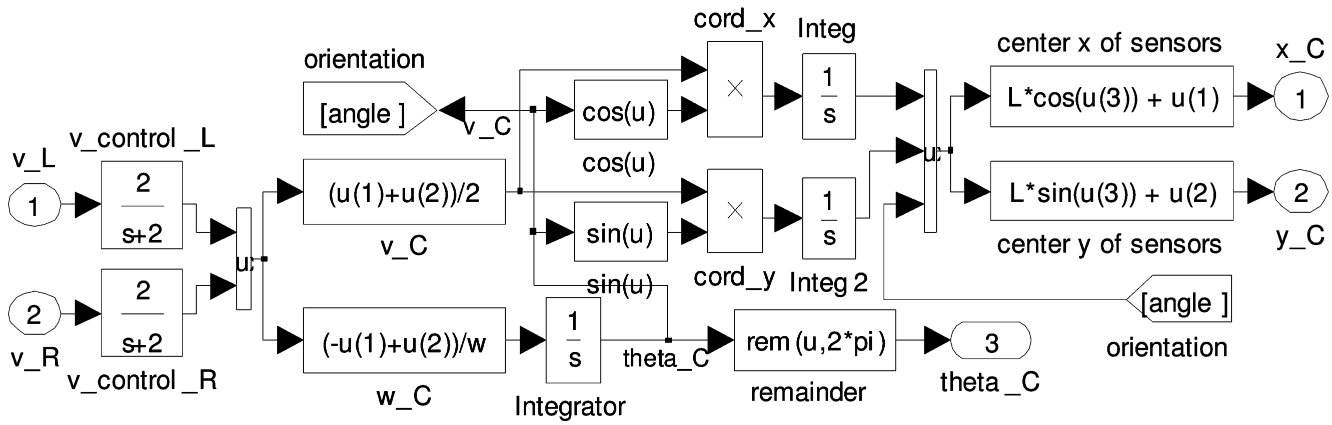


Fig. 9. The SIMULINK behavior model for line-following robots.

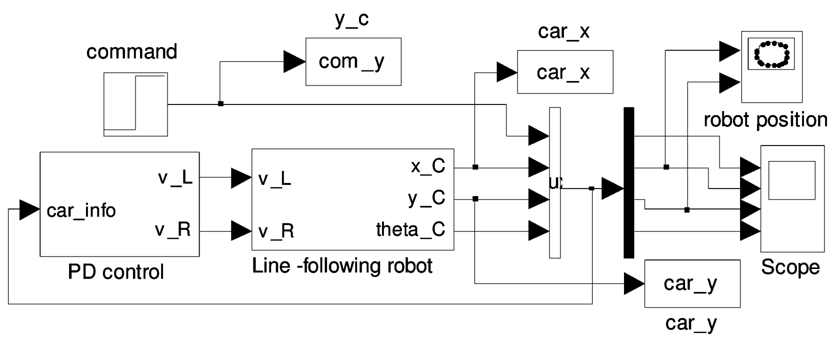


Fig. 10. The SIMULINK behavior model for PD control of line-following robots.

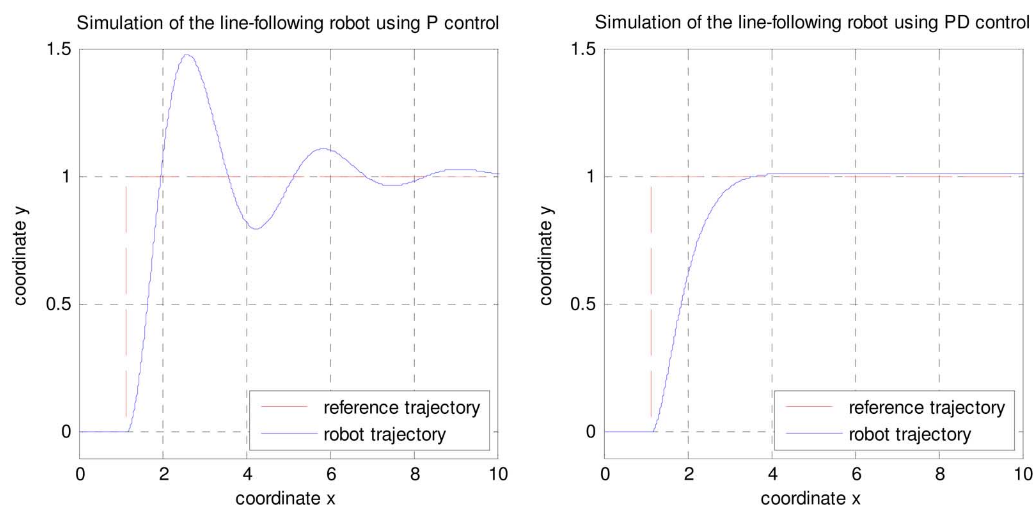


Fig. 11. The simulation results of a line-following robot using a P and PD controller.

III. Overview of the Sample Line-Following Robot

To speed up the learning process, the prototype low-cost line-following robot shown in Fig. 12 is introduced to the students. Because the C-compilers and microcontrollers and the printed circuit boards (PCBs) are supplied by Microchip Inc. and the university, respectively, the students have to spend only about $65 USD (see Table I for cost analysis) to make their own line-following robots. After all the hands-on laboratory exercises are finished, the students are well prepared for both the software algorithms and hardware circuits used in designing and building their own line-following robots.

The sample line-following robot, whose block diagram is shown in Fig. 13, consists of a microcontroller dsPIC30F4011, dual H-bridge dc-motor drivers, an RS232 port to communicate with a PC, an LCD display for debugging, an in-system programming (ISP) port for downloading programs from a PC to the 48-Kb flash memory, 9 A/D ports for accessing the sound signal, the reflective optical sensor circuit signals, and a microphone to detect the sound of a whistle in the racing contest.

The interrupt-driven real-time firmware flowchart is also shown in Fig. 14, whose sampling rate can be as fast as 1 kHz.

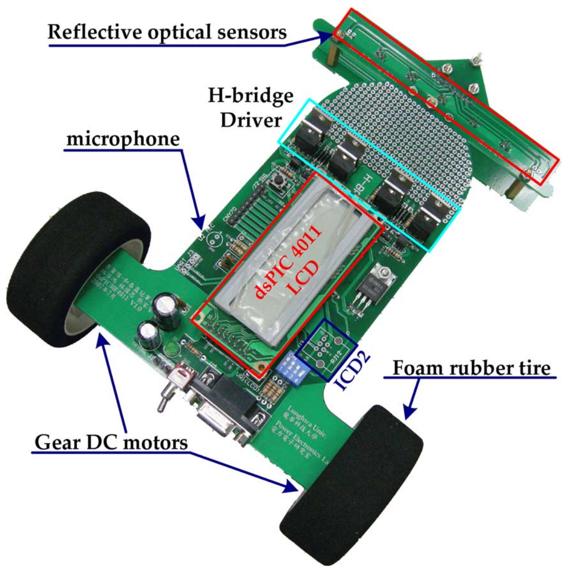


Fig. 12. The hardware of the sample line-following robot.

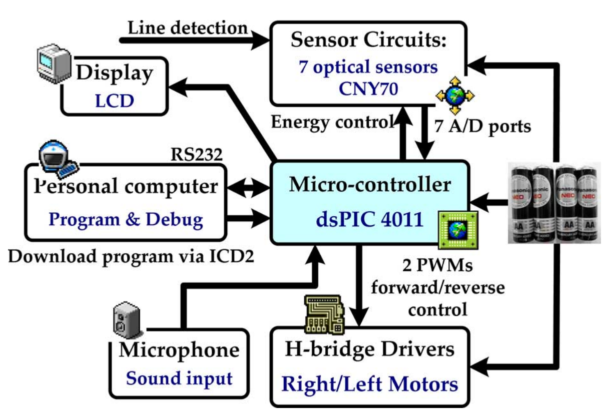


Fig. 13. The block diagram of the line-following robot.

TABLE I
COST ANALYSIS OF THE LINE-FOLLOWING ROBOT

Components and subsystems	**$USD**
LCD display	3
7 Reflective optical sensors (CNY70)	6
ISP, Power circuit and rechargeable batteries	15
Dual H-bridge converter circuits	6
2 DC gear motors + crutches + tires	35
Total	65

In completing the final project of designing and building fast line-following robots, the students would also learn that the choice of sampling rate should be based on the following considerations.

- *Energy consumption*: The energy consumed in reflective optical sensor circuits is proportional to the sampling rate. This energy-consumption relationship poses a problem, especially when the contest for line-following robots restricts the contestants to use only disposable batteries with poor discharging characteristics [8].

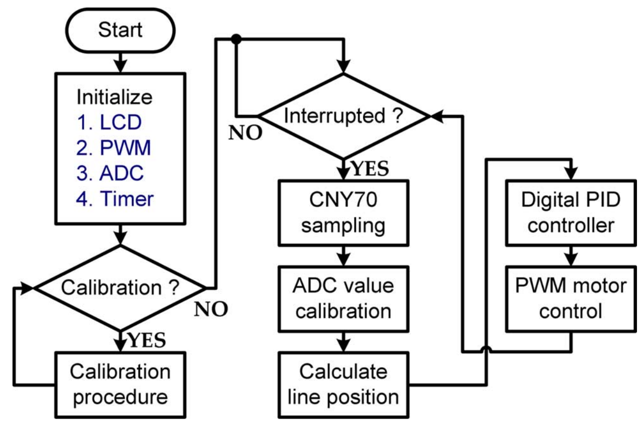


Fig. 14. The interrupt-driven firmware flowchart of the line-following robot.

- *The bandwidth of the reflective optical sensor circuits*: The analog output of the circuit in Fig. 2 may take as long as 700 μs to reach a steady state.
- *The bandwidth of motor speed control*: The motor speed may not react to high frequency variations of commands.
- *The execution time of the digital control algorithm*: This determines how fast the commands for the motors can be generated once the analog outputs of the reflective optical sensors are obtained.

The sample line-following robot constructed using the above concepts was entered in a national racing contest [8], where it won first prize and broke the existing course record of 8 s. This history of success had the effect of making the students more interested in learning the design and implementation skills included in this laboratory for embedded systems.

IV. The Assessment, Racing Contest, and Feedback

The project-based laboratory for embedded system design, which evolved from a previous course in systems design with digital signal processors, was offered to 20 senior students as a one-semester, 18-week, three-credit course in 2007. The course was organized into a one-hour lecture and two-hour laboratory each week. In every laboratory section, each group of students was assessed on their individual reports and answers to the teacher's oral questions. In their individual reports, students were required to show the results and their understanding of a particular laboratory exercise. The students also had to answer the teacher's oral questions about the firmware programming, how they finished the laboratory exercise, and what they learned in doing the laboratory exercise. This assessment scheme was fair and comprehensive for the students and prevented possible plagiarism.

To help students who did not follow the schedule very well, the teacher chose some students who did the exercises well and trained them to share their experiences during the implementation. For example, the principles of sensor calibrations and how they were implemented were a little bit difficult for some students to understand. The teacher had to make sure at least one group of students knew why the proposed method was effective. This group of students could later share their knowledge through group discussions among students.

Fig. 15. The first competitor in the line-following race.

To see how well students integrate all the skills learned in the laboratory in implementing line-following robots, a race was held after the project-based hands-on laboratory was finished (shown in Fig. 15). The course for the race is about 20 m long, and the minimum radius of curvature for the course is 15 cm. There were 12 teams that completed the racing contest, and the winner finished the race (by completing the course three times) in 44.13 s. The average speed for the winner using the sample line-following robot in Fig. 12 was about 1.36 m/s.

It is interesting to note that the racing contest did motivate the students to strive to learn the knowledge and skills necessary to make a fast line-following robot. For example, one group of students 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 Fig. 16) that could run at a maximum speed of 1.2 m/s, which is a competitive design to those devised in [17]. To measure the running time for each line-following robot, the students also devised a timer, shown in Fig. 15, that uses proximity optical sensors.

A survey was also conducted on this project-based hands-on laboratory at the end of the laboratory. It can be seen from Table II 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 were three students who thought that the cost of \$65 USD is still too high for them, which is partly due to the fact that most of the department's students are economically disadvantaged. Therefore, an even lower-cost line-following robot ($\sim$\$40 USD) will be devised for future offerings of the laboratory.

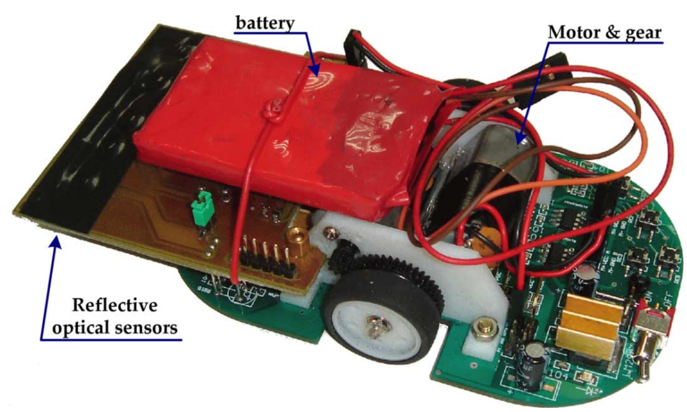


Fig. 16. The new line-following robot devised by students.

V. Conclusion

A project-based hands-on laboratory for embedded system design has been presented in this paper. The laboratory is divided into two parts. The first part helps students learn the basics of hardware circuits and software control algorithms for line-following robots, which includes introductions to the integrated development environment for microcontrollers, the interrupt-driven programming for task scheduling, the calibration procedure of sensor outputs, the quadratic line-detection algorithm, the PWM speed control strategy for dc motors, and the tracking control algorithm, behavior model simulation, and implementation. The second part of the laboratory asks the en-

TABLE II
AVERAGE SCORES OF THE SURVEY FOR THE HANDS-ON LABORATORY

Questions	Average (1-5)
The hands-on laboratory can effectively help me learn hardware circuit and programming implementation skills.	**4.06**
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**
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**
I am willing to pay the money to own the mobile robot.	**3.52**
I am satisfied with the project-based hands-on laboratory.	**4.50**

rolled students to construct line-following robots as examples of embedded system by using the mentioned techniques. The students are motivated predominantly to learn and understand the materials provided in the laboratory. The feedback of the students is quite positive, and the hands-on laboratory is also supported by the local branch of Microchip Inc., which provided the necessary C-compilers and microcontrollers free of charge.

ACKNOWLEDGMENT

The authors would like to thank Microchip Inc. for its support of free C-compilers and microcontrollers.

REFERENCES

[1] W. Wolf and J. Madsen, "Embedded systems education for the future," *Proc. IEEE*, vol. 88, no. 1, pp. 23–30, Jan. 2000.
[2] M. Moallem, "A laboratory testbed for embedded computer control," *IEEE Trans. Educ.*, vol. 47, no. 3, pp. 340–347, Aug. 2004.
[3] J. W. Bruce, J. C. Harden, and R. B. Reese, "Cooperative and progressive design experience for embedded systems," *IEEE Trans. Educ.*, vol. 47, no. 1, pp. 83–92, Feb. 2004.
[4] K. G. Ricks and D. J. Jackson, "A case for the VMEbus architecture in embedded systems education," *IEEE Trans. Educ.*, vol. 49, no. 3, pp. 332–345, Aug. 2006.
[5] S. Hussmann and D. Jensen, "Crazy car race contest: Multicourse design curricula in embedded system design," *IEEE Trans. Educ.*, vol. 50, no. 1, pp. 61–67, Feb. 2007.
[6] R. H. Chu, D. D.-C. Lu, and S. Sathiakumar, "Project-based lab teaching for power electronics and drives," *IEEE Trans. Educ.*, vol. 51, no. 1, pp. 108–113, Feb. 2008.
[7] ChaN, "Desktop line-following robot," Accessed Oct. 28, 2008 [Online]. Available: http://elm-chan.org/works/ltc/report.html
[8] "The line-following robot racing contest 2007," Fortune Institute of Technol., Accessed Oct. 28, 2008 [Online]. Available: http://www.ee.fjtc.edu.tw/contest.htm
[9] "Robotrace," New Technol. Found., Accessed Oct. 28, 2008 [Online]. Available: http://www.robomedia.org/directory/jp/game/rt.html
[10] "dsPIC30F4011/4012 datasheet," Microchip Technol., Inc., 2007.
[11] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1997.
[12] "CNY70—Reflective optical sensor with transistor output," Vishay Semiconductor GmbH, Rev. A4, Apr. 2000 [Online]. Available: http://www.vishay.com/docs/83751/cny70.pdf
[13] D. Cook, "Sandwich, an easy to build robot that follows lines," ROBOT ROOM, Accessed Oct. 28, 2008 [Online]. Available: http://www.robotroom.com/Sandwich.html
[14] M. H. Rashid, *Power Electronics, Circuits, Devices and Applications*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1993.
[15] , B. K. Bose, Ed., *Power Electronics and Variable Frequency Drives*. New York: IEEE Press, 1996.
[16] A. R. Munoz and T. A. Lipo, "On-line dead-time compensation technique for open-loop PWM-VSI drives," *IEEE Trans. Power Electron.*, vol. 14, no. 4, pp. 683–689, Jul. 1999.
[17] D. Cook, "Jet—The ultra-fast line-following robot," ROBOT ROOM, Accessed Oct. 28, 2008 [Online]. Available: http://www.robotroom.com/Jet.html

Chyi-Shyong Lee received the B.S. and M.S. degrees in electrical engineering from National Taipei University of Technology, Taiwan, and the National Tsing Hua University, Taiwan, in 1979 and 1985, respectively.

From 1985 to 1988, he served as a Lecturer in the Hwa Hsia Institute of Technology, Taiwan. He was a Lecturer in the Department of Electronic Engineering, Lunghwa University of Science and Technology, Taiwan, from 1989 to 2007 and is now an Assistant Professor. Currently, his research interests include digital control of power electronic systems and the applications of microcontroller and embedded systems.

Juing-Huei Su (S'87–M'93–SM'08) was born in Tainan, Taiwan on February 17, 1965. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the National Taiwan University, Taipei, in 1987, 1989, and 1993, respectively.

From 1993 to 1995, he served as a Military Officer in the army. In 1995, he was a Senior Engineer in the Taian Electric Company, Ltd., Taiwan. Since 1996, he has been an Associate Professor in the Department of Electronic Engineering, Lunghwa University of Science and Technology, Taiwan. Currently, he is a Professor and the Dean of Student Affairs. His research interests include robust control theory, power electronic systems, and embedded control system implementations.

Kuo-En Lin was born in Taipei, Taiwan on October 10, 1984. He received the B.S. degree in electronic engineering from Lunghwa University of Science and Technology, Taiwan, in 2007. Currently, he is pursuing the M.S. degree in the same department. His research interests include micromouse implementation and the applications of microcontroller and embedded control systems.

Jia-Hao Chang was born in Yi-Lan, Taiwan on December 13, 1983. He received the B.S. degree in electrical engineering from the Lunghwa University of Science and Technology, Taiwan, in 2007. Currently, he is pursuing the M.S. degree in the same department. His research interests include home automation systems and the applications of microcontroller and embedded control systems.

Gu-Hong Lin was born in Taipei, Taiwan on August 18, 1985. He received the B.S. degree in electronic engineering from Lunghwa University of Science and Technology, Taiwan, in 2007. Currently, he is pursuing the M.S. degree in the same department. His research interests include implementation of fast line-following robots and the applications of microcontroller and embedded control systems.