A Tool for Self-Learning Assembly Language Programming and Computer Architecture: Design and Evaluation

CHI-FAI CHAU, YU-FAI FUNG

Department of Electrical Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Received 2 September 2008; accepted 24 November 2008

ABSTRACT: This article presents the design of a system that can be used in different teaching and learning activities, in particular teaching assembly language programming and basic computer architecture for undergraduate studies. The system can be a self-learning tool for students. On the other hand, teachers can utilize this system to develop prototype systems and to demonstrate theories pertaining to operations of microprocessor-based systems. This educational tool includes hardware and software components, which were developed by the authors. The design of this system is based on principles of constructivism and learning is through hands-on experience. The system is very flexible and users can incorporate new modules for more advanced learning. Due to its flexibility and ease of use, it can also be employed as a supplementary tool for teaching other engineering subjects. Data collected from our evaluation study shows that majority of users found the system easy to use. In this article, features of the system, as well as its design and evaluation results are presented. Comput Appl Eng Educ; Published online in Wiley InterScience (www.interscience.wiley.com); DOI 10.1002/cae.20310

Keywords: assembly language programming; computer-aided learning; microprocessor-based systems

INTRODUCTION

Basic computer system engineering is a fundamental topic being taught in the Electrical Engineering degree program of The Hong Kong Polytechnic University because computer systems are becoming an integral part of every electrical engineering applications. Knowledge learnt from this subject can be applied when students implement their final-year project, which usually involves tasks such as writing programs with assembly language and developing a

Correspondence to Y.-F. Fung (eeyffung@polyu.edu.hk).

microprocessor/microcontroller based hardware system. In Hong Kong, the duration of engineering undergraduate programs is still 3 years. The final-year project is a year-3 component, which is in the form of learning by doing [1,2]. Students are first introduced to the subject Computer System Engineering in year-2. Their knowledge is built by basically through memorizing the facts. In addition, not all the topics are covered during a 14-week semester; therefore, students have difficulties applying their knowledge into practice. In summary, they simply lack experience and theory/practice link to overcome problems that they encounter during their projects. In order to overcome this weakness and to help the students, we would like to implement a system in which students can learn both software and hardware techniques relevant to computer architecture themselves. This is the primary motivation behind this project.

Various tools can be used to transmit and acquire knowledge in scientific disciplines as shown in many research on this issue [3–6]. Appropriately designed educational software can act as a catalyst in the learning and teaching of all subjects, providing promising opportunities for the construction of new meanings that can be applied to the whole learning process [7,8]. For example, Moreno et al. [9] presented a software system for the teaching of superscalar and VLIW processors and Rowe and Gregor [10] described a World Wide Web based computer-based learning system. Our tool includes hardware components and a software program that enables students to implement different microcontroller-based systems, from a very basic systems such controlling the speed of a motor to the complex ones such as mobile robot navigation.

The learning model in our system is based on constructivism and students can select to study the type of information they consider appropriate by solving real problems [7,11] and through these experiences, students are able to construct their own knowledge and eventually solve problems pertaining to their own projects. In addition, teachers or developers can also make use of the system to demonstrate different topics in areas of hardware design and assembly language programming.

The article is organized as follows: in second section, we describe the system model and structure. Both hardware and software components will be discussed and applications of the system in teaching and learning activities are given in third section. Fourth section includes preliminary evaluation results and an example demonstrating the learning outcome will be presented. Finally, conclusions are given in fifth section.

SYSTEM MODEL AND IMPLEMENTATION

A typical computer-based project involves both hardware and software developments. The hardware may include sensors and actuators, while the software is used to control these hardware modules. For example, to implement a robot that can avoid obstacles when it moves, we can apply Infra-red (IR) sensors to detect obstacles and motors to drive the robot. Then the software written should monitor the signals received from the IR sensors and then produce a proper output to control the motors. Referring to this simple processor-based control mechanism and specifications of projects developed by students in the past, we have derived a system model as illustrated in Figure 1. In this model, system includes three hardware components namely input devices, output devices, and a processing unit, which is a microcontroller. Microcontrollers, such as 8051 series, Basic Stamp, or BasicX, are commonly used instead of the traditional microprocessors because of their low-cost and simple architecture. The basic operating mechanism of the system is to process input signals coming from input devices and then output a proper control signal to various output devices according to the program written by user. From our observations, such model maps well to most projects implemented by our final-year students. Since our system includes both hardware and software components and most importantly these two are closely related; therefore we name it Hardware–Software Co-design (HSC) system.

Hardware Modules

In order to enable students to learn by implementing real systems, suitable input and output devices together with a microcontroller module are provided so that they can develop projects that can tackle practical problems. The processing unit used in the

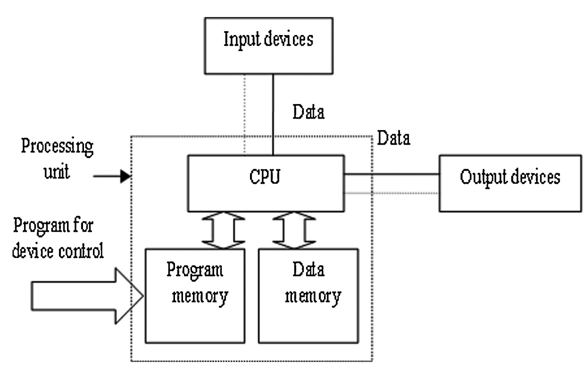

Figure 1 Model of a computer-based system.

system is the Analog Device ADuC832 processor [12]. The ADuC832 processor is already part of the experimental setup adopted in the second-year subject Computer System Principles; therefore, students are all familiar with features of the device. The basic architecture of the ADuC832 processor is based on the 8052 microcontroller. Its basic features such as 8-bit I/O ports, timers as well as the instruction set are all compatible with 8052 microcontroller. In addition, the processor also includes many useful features such as 12-bit Analog to Digital Converters (ADCs), Digital to Analog Converter (DAC), and Pulse Width Modulation (PWM) outputs. Many students are utilizing this device to implement their projects due to its versatility.

After evaluating many projects implemented by students, we were able to identify commonly used input and output devices to be included in the system. For input, it includes IR sensors, keypad, and touch switches. The output modules include a robotic arm, motor and wheel set, a speaker, and seven-segment displays. By combining different input and output devices, systems such as an electronic organ, an alarm system, or a robotic car, as shown in Figure 2, can be implemented. The robotic car makes use of the motor and wheel set, the robotic arm and the keypad. By using different keys in the keypad, actions such as pickup an object, moving in different directions can be performed.

The hardware modules are interacting with each other by connecting to the I/O ports of the ADuC832 processor. All input and output devices are implemented as modules; therefore, we can develop new modules and making the system flexible. Most importantly, students can also adopt the modules to their own projects; an example of such is described in fourth section.

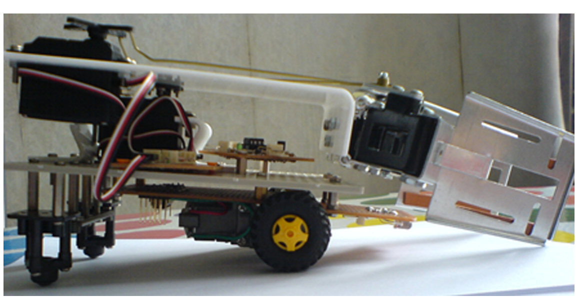

Figure 2 A robotic car implemented with different input and output modules. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

The Software Component—Motivating Combinable Assembly (MCA)

After connecting the hardware modules together, the next and also the most important step is to write a proper program. For our students, this is a difficult task, for reasons discussed in the Introduction Section. The major role of the software component, which is called *M*otivating *C*ombinable *A*ssembly (**MCA**) is to assist students to develop a proper assembly language program for the ADuC832 processor so that control of the input/output devices can be achieved. The following design criteria are considered:

(I) The system must be user-friendly and include proper instructional information to guide the users. A graphical user interface (GUI) based on the Windows Forms programming methodology [13] is implemented. Windows Forms programming is easy to implement and maintain. A prototype system implemented with the Windows Forms was used to test the effectiveness of the GUI based on the style of forms and according to the collected feedback, more than 70% of the users were satisfied with the form-based operation environment.

(II) As a self-learning tool, the system must have an easy to follow procedure and a top−down approach is adopted for configuring the program.

(III) The system must support the hardware modules. For example, if the hardware system includes an IR sensor then proper software module must be provided so that users can invoke the proper software module representing the IR sensor.

(IV) In order to achieve the self-learning objective, the system should generate a program in assembly language so that users can correlate actions performed by the system to the program source codes. Source program is in the form of assembly language because students have the basic training of its syntax. In addition, assembly language program can clearly illustrate operations performed by the processor. This enables students to have a better understanding of the working principles of the system. Techniques applied in assembly language programming are similar regardless of the microprocessor type; therefore, by learning the assembly language for the ADuC832, students can easily adapt to other processors.

Structure of the MCA

The overall structure of the MCA system presented in the form of a data flow diagram (DFD) based on symbols defined by Gane and Sarsen [14] is shown in Figure 3. The Windows Forms programming is based on Object-Oriented Programming (OOP); therefore, the MCA system is also an OOP program implemented with the C++ programming language. As shown in Figure 3, there are three major processing modules namely ModGen, FrontEnd, and ProgGen. The ModGen and the FrontEnd are user interfaces for two groups of users, in general, students and developers. Developers refer to teachers or instructors and they can utilize functions provided by the ModGen to create software objects for the corresponding hardware modules rapidly with some basic programming knowledge. All software objects are grouped in a repository (S1) as shown in Figure 3 and the FrontEnd will search the repository to obtain information of the available software modules. If an instructor wants to introduce a new hardware module or concept then with the help of the ModGen module, corresponding software object can be generated and readily available for users. Since the focus of this article is on the teaching and learning aspects of the system; therefore, details regarding the ModGen module will not be discussed in details.

Students can utilize the software to develop programs for their projects and the FrontEnd provides a form-based GUI to guide students when developing a program, as shown in Figures 4 and 5. The main form, as shown in Figure 4, employs a top−down approach to guide users to complete the programming task. According to the system model, as shown in Figure 1, steps included to define a program are:

(I) Selecting the input devices.
(II) Selecting the output devices.
(III) Defining default actions performed by an output device, this step is optional as there may not be default actions to perform.
(IV) Defining actions performed by an output device with respect to status of an input device.
(V) Generate an assembly language program.
(VI) Download the program to the microcontroller.

Following the above steps, system will generate an assembly language program for the target processor. By studying the program and observing the actions performed by the system, students can learn the programming techniques required to control the hardware modules.

In Figure 5, the GUI for configuring the output devices is depicted. At this stage, a user only needs to select the hardware modules included in the computer system and determine the I/O port and pin location(s) occupied by the devices. Once a device is configured, the information will be shown in the right-hand-side column of the form. Functions that can be performed by the device are also included in the form to assist the users. When both input and output devices have been configured then users can determine actions to be performed by the output devices with respect to different input status and this is achieved by the GUI shown in Figure 6.

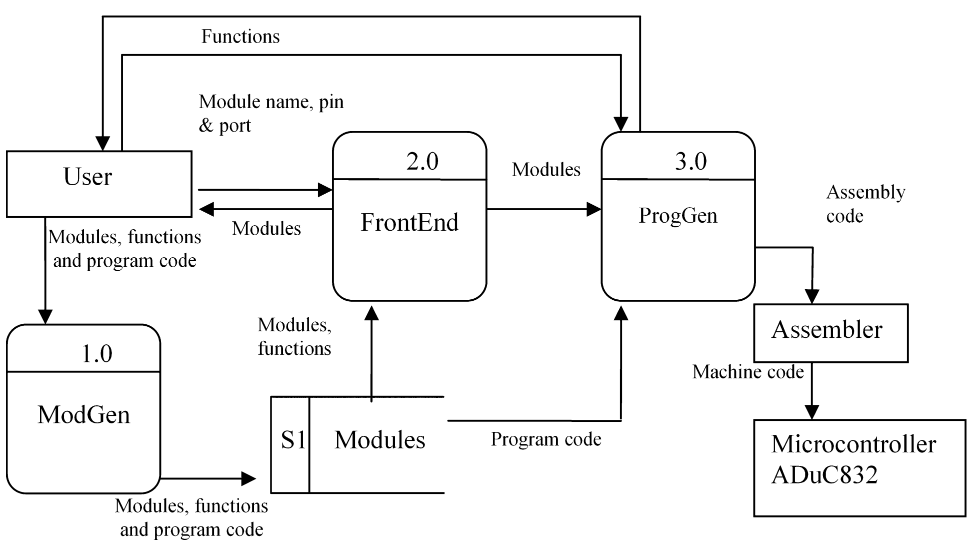

Figure 3 Overall software structure of the MCA.

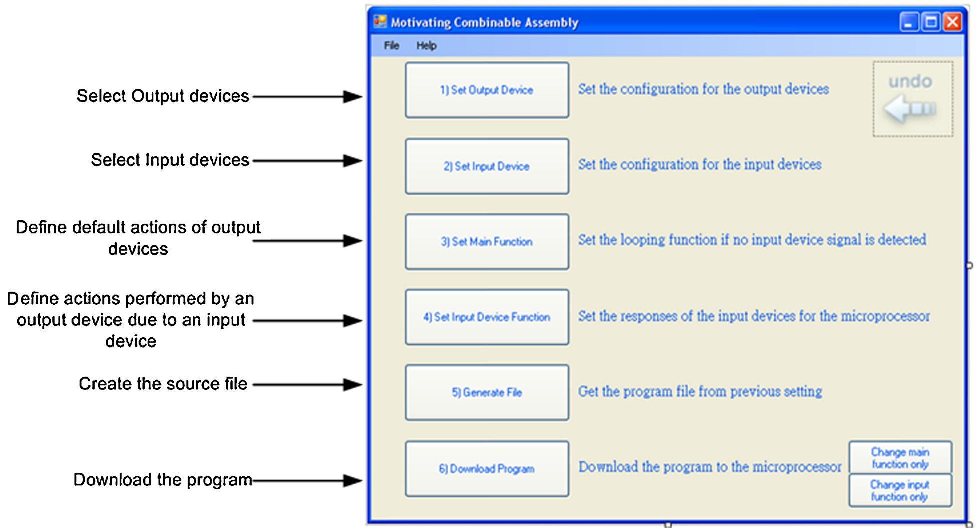

Figure 4 Main form of the FrontEnd. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

The user will first select the input device listed in the form as shown in Figure 6a. Since each input device can produce different signals, for example, an IR sensor can have the ON state or OFF state; therefore, users must also choose the desired signal to activate the output device. The device presented in Figure 6a is the keypad; therefore keys, from 1 to 9, are provided as viable input items. Columns on the right-hand-side illustrate operations controlled by the corresponding device. After choosing the input device, through the form shown in Figure 6b, users can select the output device to be controlled. The form, Figure 6b, includes the available output devices as well as functions supported by each device. The most special feature incorporated in this form is the program codes shown on the right-side-hand. This enables users to relate the program codes to the operation performed by the device and through this process they can learn the programming techniques for utilizing different hardware modules.

When both input and output devices have been properly configured, an assembly language program for the ADuC832 will be generated, to supplement the program a file describing tasks to be performed by the system is also created as a reference. Figure 7a depicts a partial program and the task description file, Figure 7b, created by the MCA system. The user can now download the program to the microprocessor and observe the system in real action. As shown in Figure 7b, we can observe how different keys of the keypad and an IR sensor can control the robotic car. For example, when the key ‘2’ is pressed, the robot

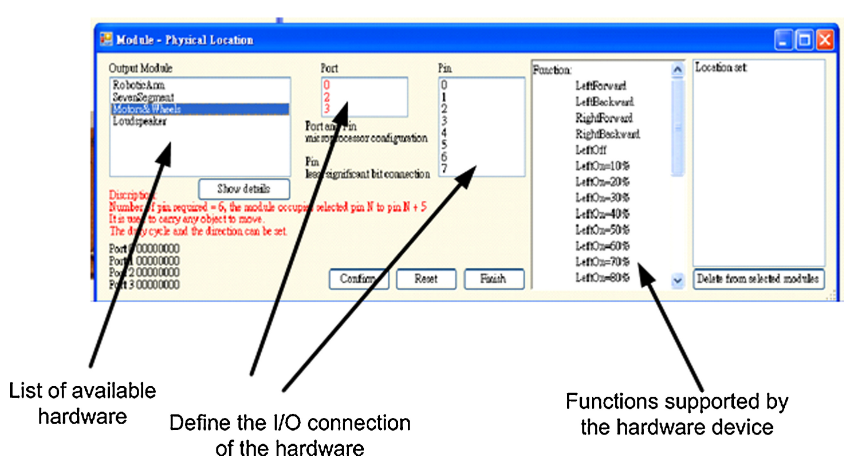

Figure 5 Form for defining output modules. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

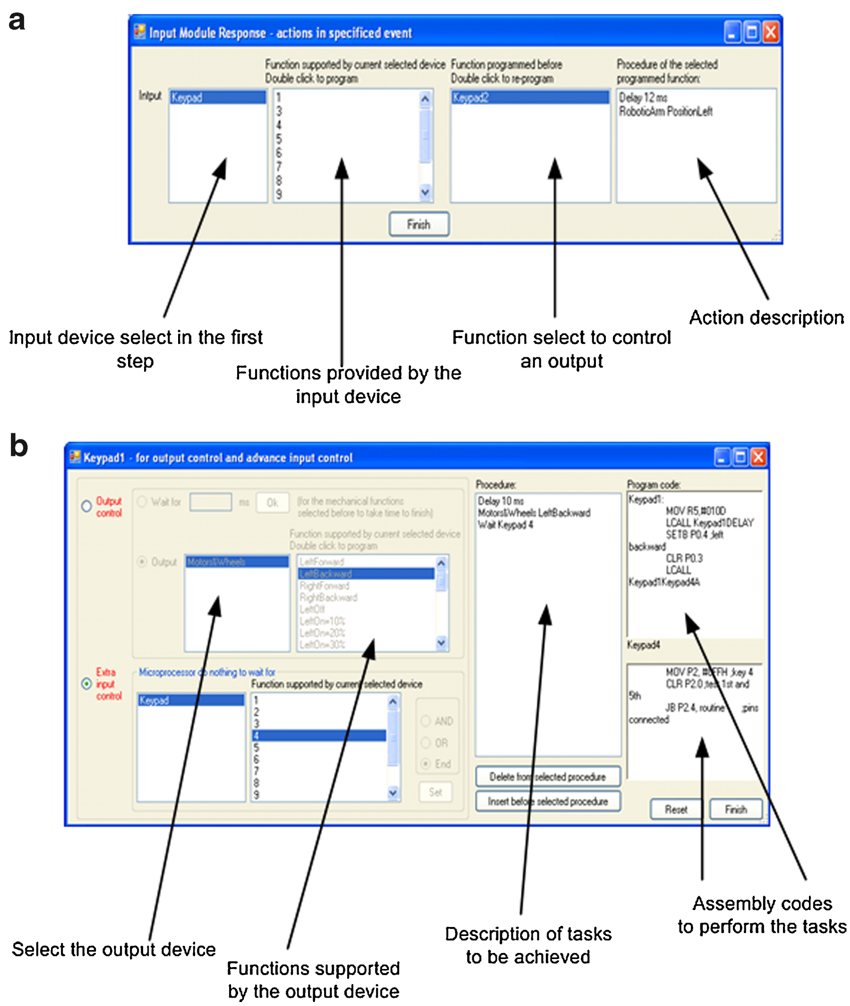

Figure 6 (a) Form to select input device. (b) Form for selecting function performed by output devices. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

will move forward and when the IR sensor is active (ON), the robotic arm will be closed to capture an object. By examining the files, the assembly language program and actions of the robot, users can gain the knowledge of hardware control using a microprocessor.

APPLICATIONS OF THE HSC SYSTEM

In second section, architecture of the HSC system has been described in details and in this section, applications of the HSC system in teaching and learning will be discussed. The major objective of the system is to allow students to learn microprocessor related system design as well as the control mechanism using assembly language programming. Based on the available hardware modules, users can design different systems and utilizing the MCA system to produce the proper control program swiftly. Since both hardware and software are presented to users therefore they can observe operations of the system. Moreover, by modifying the program, users can test their understanding of the system. Hence, by utilizing the available hardware modules, one can demonstrate different topics in hardware design and assembly language programming. For example, the structure and design of an IR sensor can easily be explained by implementing a mini-project using an IR sensor to detect the presence of an object. Using an IR sensor as the input module and the robotic arm as an output module, the control software created by the MCA system can easily achieve the task of grabbing an object when the object is sensed by the IR sensor.

Teachers can also utilize the system to develop prototype systems to demonstrate theories related to

a

```
003B                    Start:
003B 7D0A                      MOV R5,#010D
003D 1142                      ACALL DELAY
003F 02003B                    LJMP Start
0042                    DELAY:
0042 758CFA                    MOV TH0, #0FAH
0045 758A89                    MOV TL0, #089H
0048 D28C                      SETB TR0
004A                    DLY0:
004A 75A0FF                    MOV P2, #0FFH ;key 2
004D C2A2                      CLR P2.2        ;test 2nd and 4th
004F 20A403                    JB P2.4, BackKeypad2    ;pins connected
0052 12008C                    LCALL Keypad2
0055                    BackKeypad2:
0055 75A0FF                    MOV P2, #0FFH ;key 5
0058 C2A2                      CLR P2.2        ;test 2nd and 5th
```

b

```
Input device function:

Keypad2:
Motors&Wheels LeftForward
Motors&Wheels RightForward

Keypad5:
Motors&Wheels LeftBackward
Motors&Wheels RightBackward

IRSensorOn:
RoboticArm Capture

Keypad0:
Motors&Wheels LeftOff
Motors&Wheels RightOff

KeypadNum:
RoboticArm Release
```

Figure 7 (a) Extract of a source program generated by the MCA system. (b) Description file generated by the MCA system.

microprocessor-based systems. For example, basic input−output mechanism can be demonstrated using simple output devices such as LEDs or a seven-segment displays and input devices such as a keypads or IR sensors. Alternatively, a prototype system using a keypad to control a seven-segment display can be built to demonstrate basic I/O mechanism as well as operating principles of a keypad. With the program produced by the MCA, a working prototype will be available and most importantly a teacher does not need to perform a lengthy programming task. More advanced topics in microcontrollers, such as the operation of the internal timer, can be explained with practical examples. For instance, by implementing a simple DC motor control system with PWM signal. The prototype system can be developed very easily using a motor and wheel set as output and a keypad as input. With such set up an instructor can demonstrate how parameters of the timer can manipulate the speed of the motor.

A wide range of microprocessor-based devices can be implemented by putting different hardware modules together. Features of the HSC system can be enhanced further by including more hardware modules. This allows students to learn both hardware and software design and be creative with different design ideas.

SYSTEM EVALUATION

The initial system evaluation mainly focused on the system design of the FrontEnd module. Students from the year-3 class were invited to test the system and a questionnaire was prepared and results from the survey were presented in Table 1. Referring to Table 1, users in general were able to follow the procedure prescribed by the system for developing a program. The creation of the source program is an important feature and majority of the users agreed that the feature can help them to develop the software easily and more than half of the users were able to learn the programming techniques by studying the assembly program provided. Currently, the GUI is

Table 1 Results of the Survey

	Answer				
Question	0 (%)	1 (%)	2 (%)	3 (%)	4 (%)
(1) The program has a clear flow	0	8.3	33.3	50	8.3
(2) You can easily follow steps to design the software for your system	0	8.3	33.3	50	8.3
(3) The program provides clear help menu and instructional information	0	0	16.6	75	8.3
(4) You can generate a program for your computer system very easily	0	0	8.3	50	41.7
(5) The program procedures are too lengthy	0	41.7	58.3	0	0
(6) It is a valuable experience to learn by studying the source program	8.3	16.7	25	50	0
(7) The graphic user interface of the MCA is not user-friendly	0	16.7	66.7	16.7	0

Answer: 0, strongly disagree; 1, disagree; 2, no strong view; 3, agree; 4, strongly agree.

based on Windows forms, which are easy to implement and it also allows us to enforce users to follow the pre-defined programming process. Certainly, forms programming is less flexible when comparing to programs that allow users to ‘‘drag-and-drop’’ software components, such as the LabView [15] and this is certainly an area to be explored in the future.

Students who took part in the evaluation exercise were able to demonstrate constructive learning, by utilizing the system, students were able to establish a solution for a real world problem. The problem posed to them was the control of a three-phase motor. Based on the knowledge gained in controlling the motor and wheel module, students were able to learn the basic mechanism of motor control and apply it to control a three-phase motor. The block diagram of the system developed by the students is shown in Figure 8, which includes hardware modules from the HSC system and a three-phase motor. In addition to the hardware modules, the students also make use of the MCA system to first develop a prototype program to control a motor and then modify the prototype program to control the three-phase motor.

CONCLUSIONS

In this article, we described the design and evaluation of a system that supports the learning of basic computer architecture as well as assembly language programming. Our system is based on modules (both hardware and software) and therefore, developers can introduce new hardware modules to accommodate new topics to be taught. Since a computer system includes both hardware and software components; therefore, our HSC system provides modules in both forms so that users can implement real systems and observe how the software and hardware components interact.

Constructive learning is adopted and students are provided with tools to develop microprocessor-based systems and through this experience, new knowledge can be developed. Students were able to demonstrate this successfully by utilizing our system to implement a solution for a real-life problem.

System Limitations and Future Development

Currently, our system can only support hardware modules that occupy only one I/O port. Devices, such as a parallel LCD display, require more than 8-bit to control therefore cannot be included. The MCA system generates a program in assembly language according to specifications given by users but the C language is another approach for developing programs for the microcontroller-based system so we will consider implementing a different version of the software that can output a C program for the microcontroller. Our system is implemented with the Windows Forms programming methodology, which is an easy to use development tool but the form-based user interface is less attractive comparing to software that supports ‘‘drag-and-drop’’ feature. As reflected in the survey results, users prefer a more flexible GUI system. The form-based interface also

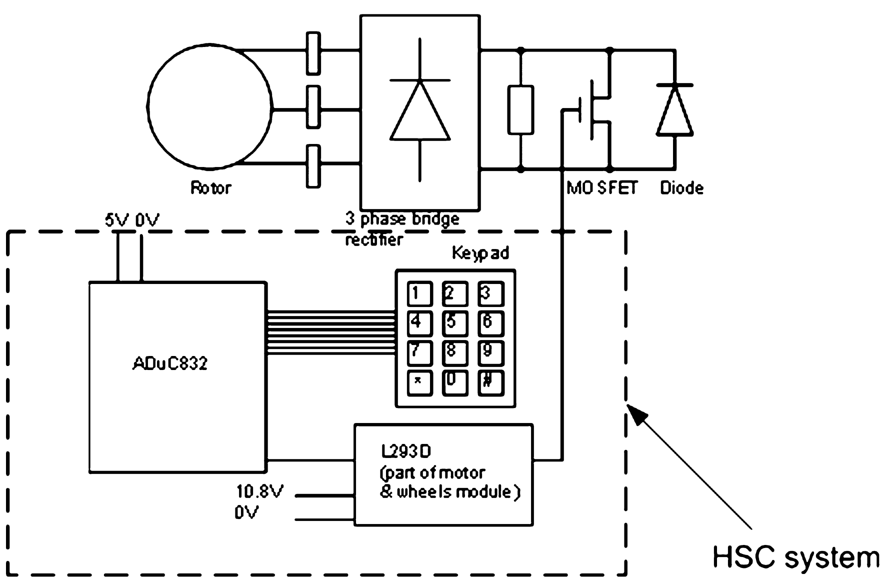

Figure 8 Application of HSC to control a three-phase motor.

limits the presentation of multiple copies of the same hardware module. For example, if the system includes four LED modules then each module is regarded as a unique component. The system developer must prepare software components for each LED module and users will be presented with four LED modules, each of which has a different name, and this may confuse the users. In addition, control that requires the combination of signals from multiple inputs is also difficult to illustrate using the form-based GUI. In future, we will try to resolve the limitations discussed above and develop a more user-friendly GUI system.

ACKNOWLEDGMENTS

The works presented in this article were supported by the Department of Electrical Engineering of The Hong Kong Polytechnic University.

REFERENCES

[1] I. Harel and S. Papert, Constructionism, Ablex Publishing, New Jersey, 1991.
[2] S. Papert, Mindstorms: Children, computers, and powerful ideas, Basic Books, New York, 1993.
[3] B. S. Eylon, M. Ronen, and U. Ganiel, Computer Simulations as tools for teaching and learning: Using a simulation environment in optics, J Sci Educ Technol 5 (1996), 93−110.
[4] F. Goldberg, Constructing physics understanding in a computer supported learning environment, Proc. of the international conference on undergraduate physics education. Part II: Sample classes, American Institute of Physics, 1997, pp 903−911.
[5] D. J. Grayson and L. C. Mc Dermott, Use of the computer for research on student thinking in Physics, Am J Phys 64 (1996), 557−565.
[6] R. N. Steinberg, Computer in teaching science: To simulate or not simulate? Phys Educ Res (A Supplement to the Am J Phys) 68 (2000), S37−S41.
[7] D. H. Jonassen, Designing constructivist learning environments, Instructional design theories and models, Lawrence Erlbaum, NJ, 1999, pp 215−239.
[8] R. Noss and C. Hoyles, Windows on mathematical meanings: Learning cultures and computers, Kluwer Academic Publishers, Dordrecht, 1996.
[9] L. Moreno, C. Gonzalez, I. Castilla, E. Gonzalez, and J. Sigut, Applying a constructivist and collaborative methodological approach in engineering education, Comput Educ 49 (2007), 891−915.
[10] G. W. A. Rowe and P. Gregor, A computer based learning system for teaching computing: Implementation and evaluation, Comput Educ 33 (1999), 65−76.
[11] T. Duffy and D. Cunningham, Constructivism: Implications for the design and delivery of instruction, Handbook of research for educational communications and technology, Macmillan Library Reference, MA, 1996, pp 170−198.
[12] Analog Devices, ADuC832 data sheet, Analog Devices, MA, 2002.
[13] H. M. Deitel, P. J. Deitel, J. P. Liperi, and C. H. Yaeger, Visual C++.Net How to program, Prentice Hall, New Jersey, 2004.
[14] C. Gane and T. Sarsen, Structured systems analysis: Tools and techniques, Prentice Hall, New Jersey, 1979.
[15] G. Faraco and L. Gabriele, Using LabVIEW for applying mathematical models in representing phenomena, Comput Educ 49 (2007), 856−872.

BIOGRAPHIES

Chi-fai Chau graduated from The Hong Kong Polytechnic University in 2008 with a first-class honors degree in electrical engineering. He is currently working at the Hongkong Electric Company Limited as an Engineering Trainee.

Yu-fai Fung received the BSc(Eng) degree in 1986 in electronic engineering with computer science, and the PhD in 1991, both from University College, London. He is currently an assistant professor in the Department of Electrical Engineering of The Hong Kong Polytechnic University. His research interests include Image Processing, Robotics, and Parallel Computing.