
A Paravirtualized Embedded System Simulation Training Scenario

Zhen Zhao ¹, Xinglu Ma ¹, Mengwei Xie ¹, Yanqing Li ¹, Jinghao Xu ¹,
Yang Gao ², Qiang Hu ^{1,*}

¹Information Science and Technology School, Qingdao University of Science and
Technology Qingdao, China;

²College of Information Science and Engineering, Ocean University of China.

*huqiang200280@163.com

Abstract

In view of the shortcomings of previous traditional experiments and training systems with high damage rate, high construction cost, the lack of real application scenarios, and full virtually experimental system can not provide real development environment. Thus this paper proposes a paravirtualized embedded simulation training. The scenario includes an embedded system development board, a virtual application scenario software, its running equipment, a virtual scene interface board, and a communication interface between the two. Using this simulation training scenario, in conjunction with the user's embedded system development board and the corresponding cross development environment, a complete embedded system training project can be constructed to ensure students' hands-on practice in the actual development environment, as well as the completion of the experimental program. After the application scenario environment is virtual, sharable and reusable, it is conducive to the actual development of students' ability to accumulate practical experience, reducing laboratory construction and maintenance costs.

Keywords

Paravirtualization, embedded system, simulation, training.

1. Analysis of Existing Problems in Traditional Experiments and Training Systems

With the development of science and technology, society's requirements for the practical ability of college students have been higher and higher, and it is becoming more and more important to train students' practical ability. Not only has the conventional experimental content been strengthened, but also experimental projects have increased, and project training has been focused on construction.

In the process of the actual embedded system, it includes (1) the core part of the embedded system (core board); (2) the peripheral interface of the embedded system; (3) the input signal conditioning board and the output signal driver board; (4) all kinds of sensors, switches, digital devices; various types of actuators, regulators and digital devices; (5) application environment these five parts. Developers can interact, program, and debug directly through embedded systems and application environments. With the traditional embedded system experiment method, it can usually include 1-4 parts, which not only lead to a large number of internal devices of the experiment box, but also are easily damaged and inconvenient for maintenance. In order to reduce the damage, some schools now start experimenting with embedded system development boards which usually include the 1 and 2 parts while a small number of boards also include 3 or 4 parts.

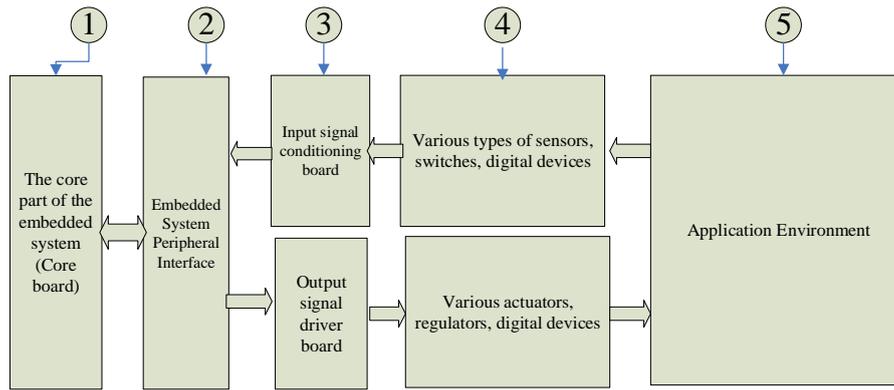


Fig 1. Key components of an embedded system

Both of the above methods do not include part 5, which results in the students being able to do some confirmatory experiments whether it is a test box or a development board. Although sometimes some comprehensive training items will be developed, without a real application environment, students' interest is not high and the effect is not ideal. Meanwhile the real application environment is difficult to set up in the laboratory. Even if it is built, because of the high cost, only a few sets can be built for demonstrations, and students cannot actually write programs to operate.

Full-virtual embedded system experimental methods can implement part 5 by using virtual simulation methods. However, the existing embedded system virtual experiment system cannot provide students with a real development and operating environment. However, the programs written in the virtual simulation environment, the development and debugging experience obtained is quite different from the actual application, which does not exercise well and improve students' actual development capabilities.

2. Paravirtualized Embedded System Simulation Training Scenario Design

By paravirtualizing the 3-5 part in Fig. 1, retaining 1-2 parts and adopting the development board approach, we propose a paravirtualized embedded system simulation training scenario (hereinafter referred to as paravirtualized scene) in the light of the high damage rate, high construction cost, and lack of real-world application scenarios for traditional experiments or training systems, the full-virtual experimental system cannot provide the real development environment and other shortcomings[1,2]. Its composition is shown as in Fig. 2, including the embedded system development board, the virtual application scenario software and its running equipment, the interface board of the virtual scene and the communication interface between the two boards. Using the simulation training scenario, with the user's embedded system development board and the corresponding cross-development environment, a complete embedded system training project can be constructed.

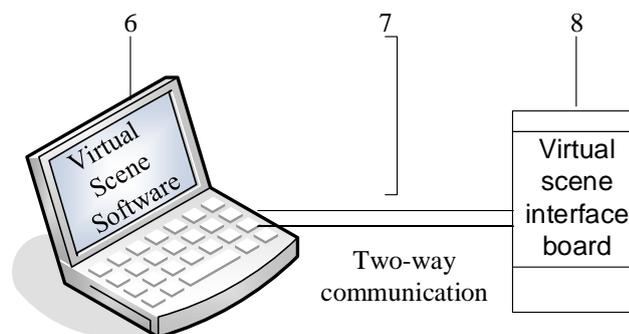


Fig 2. Par virtual embedded simulation training scenario composition

Among them, the component 6 is a virtual application scenario software and its running device, which is used to replace the 3-5 part in Fig. 1. The virtual application scenario software can acquire the data transmitted through the virtual scene interface board, and use the software to simulate and demonstrate the corresponding operation effect of the actual device (ie, the virtual device) according to the data, and pass back the necessary data during the operation of the virtual device to the virtual environment. Virtual scene interface board. The running device of the virtual application scenario software may be any devices that can run virtual application scenario software and display a virtual effect, such as a PC, a tablet, a notebook, a mobile phone, an embedded system device, and so on. Component 7 refers to the means of communication between the component 6 and component 8. The communication method may be a method to communicate based on a universal asynchronous serial communication interface protocol, including RS232, RS485, Bluetooth, infrared, ZigBee, or a physical communication interface based on a TCP/IP communication protocol, as well as a wired Ethernet communication interface, and Wireless WiFi communication interface. The component 8 is a virtual scene interface board for receiving or outputting a digital signal or an analog signal, and is matched with the component 2 in Fig. 1.

The paravirtualized scene preserves the actual development environment in the embedded system experiment, facilitates the development of the student, virtualizes only the actual application scenario, and solves the shortcomings of most current experimental or training systems.

The paravirtualized scene can guarantee students to develop and debug the experimental program under the actual programming environment, and the application scenario environment after the experimental program is completed is virtual, shareable, reusable, and extensible[3, 4]. This is conducive to training students' actual development capabilities, reducing experimental construction and maintenance costs. As application scenarios are virtual and highly scalable, they can further integrate actual project projects and virtualize the application environment of these projects. Furthermore, they can build an upgraded, new training platform consistent with the actual product project, so that students can even carry a portably embedded system development board, virtual scene interface board and virtual application software, going back to the dormitory to study. The time and places are not confined to school hours and laboratories.

3. Virtual Application Software Design

Component 6 virtual application scenario software is developed using C++ language, QT development environment, and OpenGL function library. The software interface displays a three-dimensional scene[5]. The user can drag and rotate and observe the operating status of the virtual device in multiple dimensions. The software simulates a variety of sensors and actuators internally and acts in conjunction with virtual devices. The various types of data received from the virtual scene interface board are used to drive actuator actions. After the actuator generates an action, the virtual device simulates the actual device and generates a corresponding action. At the same time, the software also simulates various signals that need to be detected during the operation of the actual device, and sends these signals to the embedded system development board through the virtual scene interface board.

Using VR technology, the part of the product development that has nothing to do with information expertise is virtualized with software, combined with specially developed interface cards, and connected with actual development equipment to construct a paravirtualized simulation training environment[6]. The environment is applied to the experimental and practical training of professional courses. As shown in Fig. 3, the platform is equipped with relevant experimental equipment, such as a single-chip microcomputer development board, which can meet comprehensive training projects such as course design.

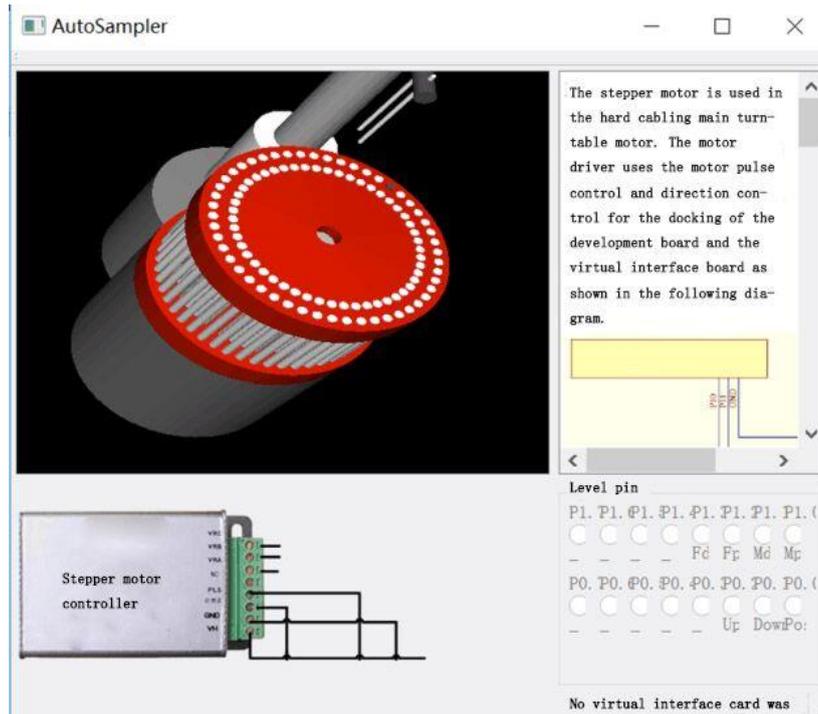


Fig 3. Par virtualized automatic sampler simulation software

4. Virtual Application Interface Board Design

The virtual scene interface board has two implementation modes: a low-speed interface card based on an embedded microprocessor; and a high-speed interface card based on a CPLD/FPGA [7]. The electrical schematic diagram of virtual scene interface board based on the embedded microprocessor-based is showed in Fig. 4.

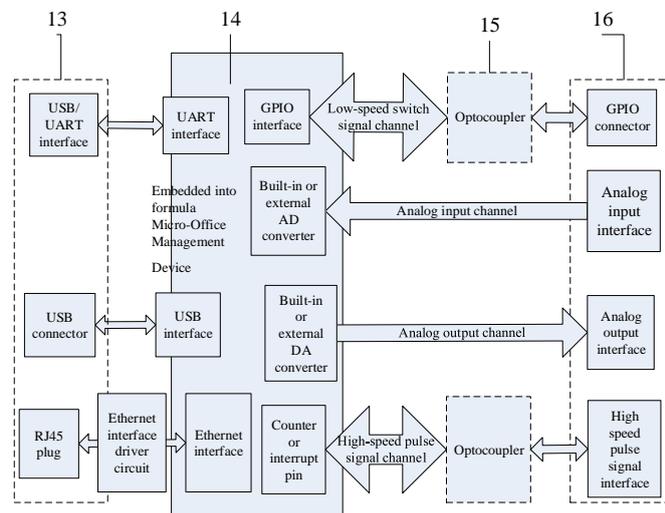


Fig. 4. Electrical schematic of the virtual interface board

In Fig. 4, the component 14 is an embedded microprocessor, which may be a single chip microcomputer or an ARM processor. According to different virtual application scenarios, analog input and output channels and converters are optional. If there is only a digital signal in the virtual application scenario, a microprocessor without AD converter and DA converter can be selected. For high-speed digital signals, such as pulsed signals, use the embedded microprocessor's counter or interrupt pin to capture or output [8].

The component 15 is added to improve the reliability of the interface board, and is used to implement electrical isolation between the interface board and the embedded development board. When the

system has an analog channel, no electrical isolation is required and component 15 can be removed. Component 16 is a generic term for various GPIO connectors, analog input/output interfaces, and high-speed pulse signal interfaces. Component 13 is a communication interface between the virtual scene interface board and the virtual installation scenario software. The UART interface, USB interface, and Ethernet interface shown in the figure can be provided at the same time, or you can have one or two of them.

5. Summary

The paravirtualization simulation system proposed in this paper provides a feasible teaching model for the cultivation of embedded system talents. It improves the traditional experiment's high cost and complexity of operation and the low realism of the full virtual experiment. Through the combination of virtualization technology and embedded development board, students get a kind of real-time effect response, and to a certain extent, reduce the cost of IT professional training. And the encapsulation features provided by virtuality make it possible to reuse, share, and extend experimental programs. Such a common practice model of hardware and software, in the enhancement of students learning interest while improving the learning efficiency, training students' practical development ability.

References

- [1] Josip Prieto-Basques, Jordi Herrera-Joancomarti, Ana-Elena Guerrero-Roldán. A Virtual Laboratory Structure for Developing Programming Labs [J]. *International Journal of Emerging Technologies in Learning*, 2009, 4(S1:+SIIE).
- [2] Dimitris Nathanael, Sergio's Moisakos, George-C. Vosniakos. Development and evaluation of a virtual training environment for on-line robot programming [J]. *International Journal of Industrial Ergonomics*, 2016, 53.
- [3] Diane Barrett. How Virtualization Happens [M]. Elsevier Inc, 2010-06-15.
- [4] Yong Feng Yin, Bin Liu, Guo Qi Li, Zhi Wang. Embedded Software Simulation Testing Virtual Machine: Design and Application [J]. *Applied Mechanics and Materials*, 2010, 973(26).
- [5] Wai-Leong Yew, Cedric Westphal, Ulas C. Kozat. Designing and embedding reliable virtual infrastructures [J]. *ACM SIGCOMM Computer Communication Review*, 2011, 41(2).
- [6] Orit Hazzan, Tami Lapidot. The practicum in computer science education [J]. *ACM SIGCSE Bulletin*, 2004, 36(4).
- [7] Chun Ming Huang, Kai Chao Yang, Yu Tsang Chang, Chien Ming Wu, Shian Wen Chen. A Tiny Development Platform with Virtualized Peripherals for Education of Embedded Software Design [J]. *Advanced Materials Research*, 2013, 2493(748).
- [8] Alexandra Aguiar, Fabiano Hessel. Current techniques and future trends in embedded system's virtualization [J]. *Software: Practice and Experience*, 2012, 42(7).