A Survey on Impact of Embedded System on Teaching

Mitali Agarwal  
ECE Department  
MIT, Moradabad, UP, INDIA  
Email: mitali.mit@gmail.com

Pankaj Bhardwaj  
ECE Department  
MIT, Moradabad, UP, INDIA  
Email: april5pankaj@gmail.com

Amit Sharma  
ECE Department  
MIT, Moradabad, UP, INDIA  
Email: amit198730@yahoo.com

Vedpal Singh  
CSE Department, Dev Bhoomi Institute of Technology (DBIT), Dehradun, UK, INDIA  
Email: vedpalsiet101@gmail.com

ABSTRACT

The paper introduces the purpose and signification of the course of embedded system, and then expounds the main teaching content of embedded system which students must learn, such as the structure of hardware, boot loader and device driver, embedded real-time operation system, debug method for embedded system, the application program of embedded system, etc. We teach the experiment of embedded system with new thought, and designed corresponding experiment. Students are required to familiar with the method of embedded system design, know about real-time operation system, master workbench for embedded system development, familiar with debug method, and understand the co-design of software and hardware. With the method, students can master the development of embedded system rapidly, and the teaching effect is well.

Keywords: Embedded system, Teaching, Experiment.

I. INTRODUCTION

The embedded industries in many countries have been a booming industry session in the last few years. Embedded systems represent a major fraction of the digital systems market as indicated by the fact that embedded systems represent a key technology in the automotive, consumer electronics, industrial automation, military and aerospace applications, office automation, telecommunication and data communication industries. As much as 98% of all 32-bit microprocessors currently in use worldwide are used in embedded systems. These embedded processors provide special purpose functionality as opposed to the general purpose applications familiar to desktop computer users. A recent report estimates that the typical household has 100 processors in its confines. The growth rate for embedded processors far exceeds that of traditional computers. For this reason, educating our current engineering students in the best practices for real-time and embedded systems development is of great importance. With the needs of embedded system engineers greatly increase, the industry have complained that neither computer science curricula nor electrical engineering curricula provide sufficient skills for junior engineers to design embedded systems. Many of real-time and embedded systems directly interact with sensors and actuators or are safety critical components within larger systems. This imposes significant system constraints with respect to response time, platform architecture and safety considerations not found in general purpose applications. The standard computing curricula concentrate primarily on general-purpose desktop applications and do not provide students with the opportunity to gain the necessary skills for engineering software in real time and embedded systems. Thus embedded systems education is an excellent example of an area of study that requires depth and rigor while maintaining breadth required for meeting emerging workforce and education needs. Recent years have seen much discussion about the appropriate curricula for embedded systems design. Many universities have offered courses related to embedded systems design. The courses can be categorized into three categories: hardware oriented, software oriented, and HW/SW integration design. Traditional Computer Science curricula focus on the training for logic reasoning and programming skills. System integration is often not covered in computer science curricula. As the embedded platforms migrate from 8-bit microprocessors to 32-bit microprocessors, the engineers require different skills to design modern embedded systems. Most computer engineering programs teach programming and design skills that are appropriate for a general-purpose computer operating
under control of a commercial operating system rather than for the more specialized embedded systems. Embedded systems courses are being integrated into computer science, electrical and computer engineering, software engineering programs, and cross-disciplinary courses. Current embedded systems courses span a number of topics including designing real-time systems, low power systems that contain reconfigurable hardware, embedded processors, software, and digital signal processing. For most of industry application, HW/SW integration design is required. To meet the requirement, the teacher should provide the fundamental concepts for real-time systems, embedded hardware, real-time scheduling, real-time operation systems, power management, In-circuit emulator, workbench, and embedded real-time programming, etc. With this class, students should learn how to design embedded real-time systems from software perspective and how the software and hardware components collaborate.

II. THE MAIN CONTENT OF EMBEDDED SYSTEM COURSE

One of the objectives for students is to be able to enter professional practice. To learn embedded system, the course must include five parts: hardware architecture, boot loader and device driver, embedded real-time operation system, debug method for embedded system, embedded application, etc. Embedded system hardware architecture. Real-time and embedded systems operate in constrained environments in which computer memory and processing power are limited. They often need to provide their services within strict time deadlines. The embedded system is hardware-dependent strictly, for a certain application, there is certain hardware structure. The main contents include embedded processor, memory, bus, interface, communication, etc. The course incorporates embedded systems concepts into the context of computer architectural issues. Traditional computing architectures are introduced, evaluated, and contrasted with embedded systems architectures. Specifically, architectural design tradeoffs associated with the processor(s), input/output (I/O), and memory are discussed. After the learning of this part, students are required to choose suitable hardware to design embedded system to meet the certain requirement. Boot loader and device driver. Boot loader will set some system settings, and then will be used to load the kernel, and transfer control to the kernel for system operation. In embedded systems particularly, boot loaders are very important because of doing work on the kernel: the kernel can be altered and tested, and the boot loader will automatically load new kernel into memory. A device driver is a computer program allowing higher-level computer programs to interact with a computer hardware device, including Board- support Package (BSP) and Hardware Abstraction Layer (HAL). A driver typically communicates with the device through the computer bus or communications subsystem to which the hardware is connected. Drivers are hardware-dependent and operating-system-specific. Embedded Real time operating system. Real-time operating system (RTOS) is a multitasking operating system intended for real-time applications. An operating system is considered real-time if it invariably enables its programs to perform tasks within specific time constraints. The RTOS performs few tasks, thus ensuring that the tasks will always be executed before the deadline. The RTOS drops or reduces certain functions when they cannot be executed within the time constraints. The RTOS monitors input consistently and in a timely manner. The main contents of RTOS include the fundamental concept, algorithm, kernel, task, interrupt and the transparent of RTOS, etc. Students are required to master the basic concept of RTOS, and transparent the RTOS to certain embedded system. In embedded software development, Students were introduced to the ideas of cross-platform development including host based embedded software development and embedded target environments, integrated development environments, interrupts and interrupt handling, and embedded software architectures. Then the course introduced the existing techniques from software engineering, and briefly covered several topics including embedded software development, software development lifecycle, software development models, including the waterfall model, the spiral model, rapid application development model, and object-oriented approaches. Universal modeling language (UML) was introduced as an important formal method for software engineering. We also concentrated on specific techniques for testing, verification and validation for embedded systems.

III. THE MAIN EXPERIMENTAL ITEMS

The embedded system course is application oriented; learning embedded system is not only helpful for students to familiar with the methodology of system design, but also improves students’ ability to master and apply correlative knowledge. For the teaching of embedded system, experiment is very important, and it is the key way for students to master the technique of embedded system design. In college or University, the course of embedded system is usually arranged on 6th or 7th semester, and is defined as an elective course. The credit hour for embedded course is about 3, after the classroom teaching, the time can do experiment is only about 10–20 hours. With the short time for experiment, students must try his best to familiar with embedded system, otherwise he will be failure to learn this course, and then the experiment item and content will be important for master embedded system. The content of the experiment should have the systemic thought in mind, and then students can do experiment from fundamental technique to complex system design, by this means, students can be familiar with the method of system design, and master development tool and verify tool. Then we teach experiment as two parts, one is fundamental knowledge and development tool, another is system design. The aim of fundamental knowledge and development tool experiment is that students can know system design method, know a kind of embedded processor, know real-time operation system, master a kind of development tool, know debug method, know software and hardware co-design. The aim of system design
experiment is that students can design a certain embedded system to meet application with the embedded experimental system. To achieve the purpose of experiment, we adopt the experiment system based on ARM processor, and design some experiment item. Fundamental knowledge and development tool experiment. By these experiments, students should learn how to use the development tool and programming language, and learn the bottom design of embedded system software and hardware. The main experiment items are as follow:

- ARM integrate development environment and debug method. The main contents include how to use ARM ADS 1.2 integrate development environment and emulation and verification method.
  - (i) ASM and C programming experiment. Include calling C program in ARM application and calling ASM program in C application.
  - (ii) Boot loader and BSP. Students should know the function of boot loader, and know the general method to design BSP.
  - (iii) Experiment of RTOS. The purpose is that students can master the transplant method and condition of RTOS under all kinds of hardware environment.

Experiment of system design. For student’s master the application of embedded system, two experiments of system design is adopted, the main experiment items are as follow:

- (i) Stepper motor control experiment. By the experiment, students will learn how to build an applied control system with embedded technology.
- (ii) LCD display experiment. By the experiment, students can know the basic method which develop portable display terminal with embedded technology.

IV. CONCLUSION

Embedded system is a practical course, experiment is the main way which students master the method of embedded system design. With reasonable experiment items, the teaching can get the purpose rapidly. The basic purpose for teaching embedded system is that students can master the method of embedded system design, know a kind of embedded processor, understand a kind of RTOS, master a kind of development tool, master debug method. Students can be familiar with the basic techniques of embedded system design by fundamental knowledge and development tool experiment, and analyze the actual requirement of certain embedded system by experiment of system design. With our course, most of students can learn to develop certain embedded system well.

REFERENCES


[10] Embedded.com—Under the Hood: Robot Guitar embeds autotuning By David Carey, TechOnline EE Times (04/22/08, 11:10:00 AM EDT) Embedded Systems Design-Embedded.com
