

Design and Analysis of Object-Oriented Embedded Device Detection Method

Lei Liu and Ting Cai

Jiangxi Industry Polytechnic College, Electronic and Information Engineering Branch

Keywords: Computer application; Embedded; Object-oriented; Default detection

Abstract. With the constant development of high-performance embedded processors, embedded equipment is widely used in various fields, such as military or civilian sector. It has become an urgent problem for manufacturers to guarantee the passing rate of various embedded equipment, but the existing ways to monitor embedded equipment have certain limitations. Improving the reliability of ways to measure embedded equipment can make the detection more efficient, automatic, standardized and easier to be operated. Meanwhile, it is quicker to position defaults to guarantee the test efficiency and comprehensiveness of detection, so there will definitely be wider demand for automatic system, which will have better application prospect. In this paper, the object-oriented thought is combined to design embedded equipment to improve the efficiency of detection and realize various functions of the system.

Introduction

Embedded technology is one of the fastest growing fields in the IT industry. Embedded system has very wide application field with its significance in industry development still growing, so embedded system has very broad application prospect. Embedded system, which is closely related to our life, has been widely applied to various industries currently, from digital products to cars and airplanes. Embedded application consists of Microcontroller (MCU) and Embedded Microprocessor (MpU). Microcontroller, also known as single-chip machine, is translated from Single Chip Computer, and it is mainly used for controlling and is the product of computer miniaturization. Embedded system and analog circuit or SOC (system on chip) composed of other functional circuits or Sip (system in package) are increasingly widely applied to mobiles, set-top boxes and other products with complex functions.

The application of embedded technology is the application of computer in all kinds of equipment and products. Since the beginning of 1980s, the single chip microcomputer, after nearly 40 years of development, the growing number of varieties, functions and constantly enhance the scope of application. In order to save development time (to Mark Time), embedded microprocessor is used in 16 bit and committed applications, and a large number of computer programs can be used directly. Software development has been programmed by assembly language, experienced a high-level language C programming, and began to turn on the Time Operating System RTOS (Real) on the programming. The hardware design of many products has been used in this design mode. In order to shorten the development time and strengthen management, the concept of platform has been gradually introduced into the process of product design and development. The development trend of embedded application software and hardware design is the development mode of the platform. In this paper, we study the detection scheme of embedded devices.

Embedded Platform Development Mode

The so-called platform development refers to implement strictly standardized, serialized and regulated design to integrate the basic hardware structure and basic software of a certain field of products within a design group, i.e., basic platform, and to optimize and screen its surrounding circuits to form a basic platform resource. When developing new products, take advantage of the underlying platform, and the platform of choice in the library-related resources. For example, application developers embedded systems network program, embedded systems design, et al.

Device driver. Of course, you can also compile and transplantation based on the Java virtual machine concept and KVM, the use JZME development of embedded system applications.

All embedded processors are based on certain infrastructure (i.e. IP core), many manufacturers produce processor, but with only a handful of IP core. With one's own IP core, a great fortune can be earned by selling the IP core. Therefore, the embedded system design applications to be developed soft / hard IP core industries, and improve the hard / soft IP core library functions. Experts library functions is generalized soft IP core, TCP / IP protocol network application software using the typical soft IP core. In the platform approach, basic software and hardware to work to resolve the platform. Common peripheral interface chip like 8255, 8250 and 8279 etc. can be hard to make a similar function IP core, software library functions can contain their drivers. The keyboard can be standardized into an 8 key, 4 key, 8xs key design, LED display can be standardized as 8 or 16 digital tube. Display and keyboard scan can be compiled into standard procedures. Embedded system applications communicate with each other practitioners can learn IP core, to build their own development platform.

Object-Oriented Method

Object-oriented is a brand new software design method that absorbs beneficial concepts and effective ways in the field of software engineering. This software design method integrates abstraction, encapsulation, inheritance and polymorphism into one, so it is easy to be modular, reusable, maintainable and expansive, etc. Based on the various advantages of object-oriented, it guides designing the method of detecting embedded equipment. In the device detection system is easy to manage each detection module, the detection method of each module packaged good, is easy to extend the function of each detection module, to improve the efficiency of detection, but also more conducive to subsequent maintenance.

Design and Realization of Multi-Task Detection Method

Overall Software Design of Default Detection. Default detection software consists of two parts: monitoring programming and functional programming. The former is the basis for ordinary work and the latter is to realize the default detection function.

The main role of fault detection is the ability to respond in a timely manner all kinds of requests from inside the device components, can effectively monitor their own fault management software, hardware and human-machine interface equipment, and other devices to exchange information, in case of system failure, and timely make the appropriate treatment.

The system's default monitor consists of knowledge base, logic program, knowledge base management system, knowledge acquisition mechanism, explain mechanism, dynamic comprehensive database and man-machine interface module, etc., as shown in Fig. 1

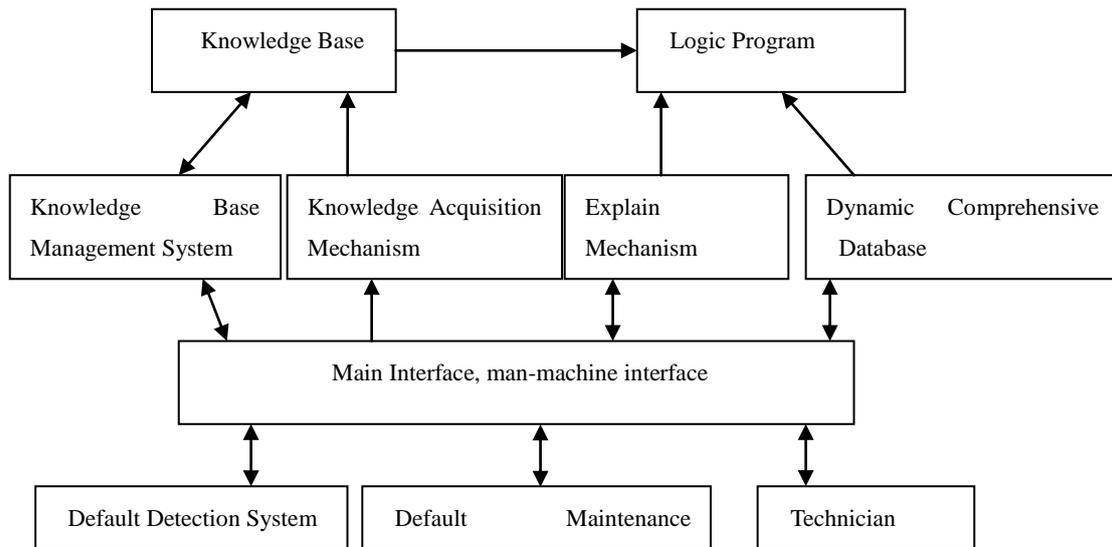


Figure 1. Computing Model of System Default Detection Software System

Main monitoring program calls for each module and associate them together to form an integral part so as to realize comprehensive management over the system. Default monitoring software in this paper mainly completes the menu operation, LCD display, data acquisition and process and PC communication protocol, etc.

Composition of Detection Module Structure. Detection module is composed of scheduling managing machine and each detection module. The scheduling managing machine is in charge of coordinating tasks of each detection module to those tasks can be carried out appropriately and tested. Besides, it gathers the detection results of each module to make corresponding judgment. Detection module's framework, principle and structure are as shown in Fig. 2

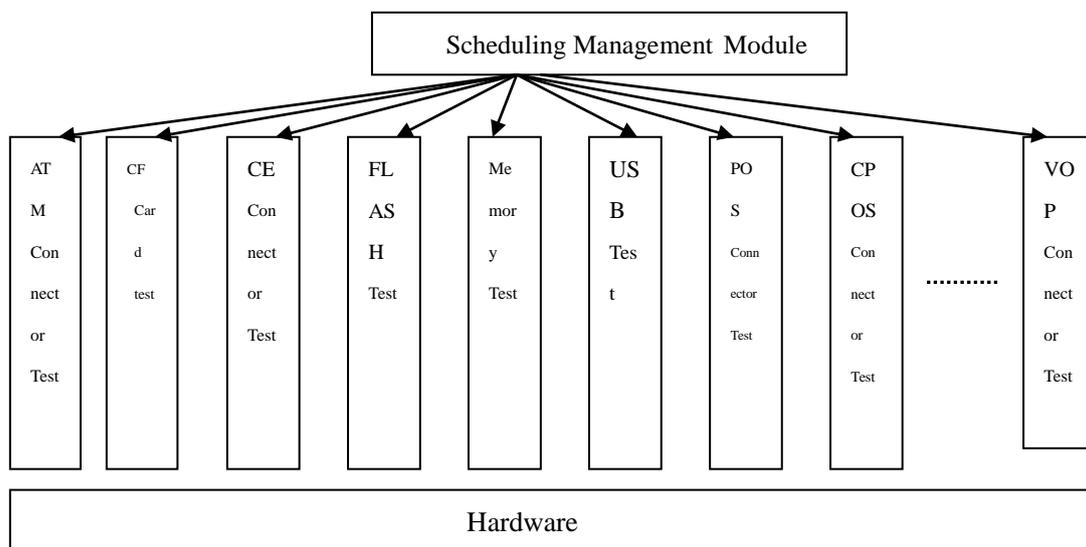


Figure 2. Detection Model Framework Diagram

Conclusion

This paper analyzes the overall characteristics and demands of large embedded equipment's detection system and demands, and designs object-oriented embedded equipment's detection system, which can manage and detect each module of equipment at the same time. Compared with non-object-oriented design method, object-oriented method has certain system and structure, which

makes it easier for maintenance, effectively reduces the iteration process and costs of developing framework as well as shortens the development circle. Meanwhile, only tiny modification can meet the detection requirement of various types of equipment and platform migration, which not only improves detection efficiency, but more importantly, improves the reliability of the system.

References

- [1] Chivers I, Sleightholme J. Object Oriented Programming [M]// Introduction to Programming with Fortran. Springer London, 2012:365-397.
- [2] Vol. N. Component Software: Beyond Object-Oriented Programming [J]. Communications of the ACM, 2015, 40(September).
- [3] Minsky N H, Rozenshtein D. A Law-Based Approach to Object-Oriented Programming. [J]. Acm Sigplan Notices, 1987, 22(12):482-493.
- [4] Drummond A, Strimmer K. PAL: an object-oriented programming library for molecular evolution and phylogenetics[J]. Bioinformatics, 2001, 17(7):662-663.
- [5] Briot, Jean Pierre, R. Guerraoui, and K. P. Lohr. "Concurrency and distribution in object-oriented programming." *Acm Computing Surveys* 30.3(1998):291-329.
- [6] Bobrow, Daniel G., and K. Kahn. "CommonLoops: merging Lisp and object-oriented programming." *Acm Sigplan Notices* 21.11(1986):17-29.
- [7] Chambers C, Ungar D. A retrospective on: "customization: optimizing compiler technology for self, a dynamically-typed object-oriented programming language"[J]. *Acm Sigplan Notices*, 2004, 39(7):295-312.
- [8] Bobrow D G, Kahn K. CommonLoops: merging Lisp and object-oriented programming [J]. *Acm Sigplan Notices*, 1986, 21(11):17-29.
- [9] Budd, Timothy A. *An Introduction to Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Inc., 2010.
- [10] Zhou, E. Z. (1996). Object-oriented programming, c++ and power system simulation. *IEEE Transactions on Power Systems*, 11(1), 206-215.
- [11] Corral, J. M. R., Balcells, A. C., Est évez, A. M., Moreno, G. J., & Ramos, M. J. F. (2014). A game-based approach to the teaching of object-oriented programming languages. *Computers & Education*, 73(1), 83-92.
- [12] Daniel G. Bobrow, Kenneth Kahn, Gregor Kiczales. 'Commonloops — merging Common Lisp and Object-Oriented Programming, [J]. *Acm Sigplan Notices*, 1986, 21(11):17-29.