コンピュータと教育　63－4
(2002. 2. 22)

A Case Study: A Fundamental Experiment Combined Hardware and Software for Information Systems Engineering

Yoshiro Imai*1, Kiyoshi Hayashi*2, Hiroshi Inomo*1, Wataru Shiraki*1, Hiroshi Ishikawa*1

*1 Faculty of Engineering, Kagawa University　*2 Dynax Takamatsu Inc.

Faculty of Engineering, Kagawa University, 2217-20 Hayashi-cho, Takamatsu , 761-0396, JAPAN
Tel: +81-87-864-2244　FAX: +81-87-864-2244　E-mail: imai@eng.kagawa-u.ac.jp

Abstract

In order to provide an efficient education to students of engineering faculties, some themes of practical engineering experiments must be discussed and developed to understand information engineering technologies related with hardware and software. We introduce an actual example of fundamental experiment combined hardware and software for information systems engineering. The contents of our working experiment are constructed with the following sub themes, namely,

1) Understanding characteristic detail and actual procedure of digital circuits by means of hardware implementation,

2) Programming of device driver for Linux and acquiring Unix manipulation environment, and

3) Applying reliability analysis into qualitative classification and quantitative evaluation for error detecting and correcting during hardware implementation and programming.

And we try to change awareness of students about experiment from that experiment will be rightfully done to that some errors will happen at experiment but it is very important to investigate what kinds of errors happen (analysis) and how errors can be avoided (improvement).

This paper describes that research backbone, conceptual idea of theme decision, details of our working experiment, and its belief evaluation.

keywords: engineering experiment, hardware implementation, Unix programming, error classification

ハードウェア製作とソフトウェア作成を融合した情報システム基礎実験

今井慈郎*1，林輝義*2，井面仁志*1，白木渡*1，石川浩*1
*1 香川大学工学部信頼性情報システム工学科，*2 (株)ダイナックス高松

〒761-0396 香川県高松市林町 2217 番 20 香川大学工学部信頼性情報システム工学科
Tel: 087-864-2244　FAX: 087-864-2244　E-mail: imai@eng.kagawa-u.ac.jp

あらまし

情報システム教育をより効果的に実施するためには，実験あるいは演習といった体験的技術修得を如何に行うかが不可避な重要課題となっている．実験テーマを検討する上で，ハードウェア製作に関するテーマおよびソフトウェア作成に関するそれは実験教育の題材として二大支柱を構成する必須アイテムである．しかし，両者を組合わせて効果的な実験教育を効率良く実施するには，様々な工夫が必要である．また，ハードウェア製作実験を実施する上では，実験実施に伴うランニングコスト発生の問題も実験担当者にとっては留意すべき重要事項となっている．最近のソフトウェア重視の傾向と相俟って，ハードウェア製作実験の比重が相対的に低下している傾向は情報システムに関する実験教育のアンバランス化を助長する遠因ともなっている．そこで，情報システムに関する基礎的事項を組み合せて，簡単なハードウェア製作からソフトウェア作成までを有機的に融合させ，情報システム構成技術の基礎を修得することを目的とした実験テーマを紹介する．テーマでは，(1)ハードウェア製作サブ実験と(2)ソフトウェア作成サブ実験とから構成される．また，(3)ヒューマンエラーや誤動作解析などエラーに関する定性分類と定量評価を実験で扱うべき主題の１つとして位置付けている．本稿では，工学実験の事例研究として，実験テーマの概要，具体的な教育方法と実験実施状況，および実験の効率化を目指す支援環境について報告したい．

キーワード：情報工学のための基礎実験，ハードウェア製作，ソフトウェア作成，誤り分類と定量評価

1. Introduction

Many types of engineering education always need their suitable themes of fundamental experiments in order to achieve their educational missions effectively and efficiently. Information systems engineering, for example, has been searching fruitful and cost-effective themes of their experiments, which are desirable to be up-to-date and continuously extensible.

Especially it is no doubt desirable that only one theme includes a lot of sub-themes from implementing hardware to programming (i.e. producing software). Almost teachers believe that repetition of fundamental experiments will make students be skillful. And suitable themes of engineering experiment, which contain both hardware and software, can lead them to be engineering experts.

It may cause a new problem how to reduce the times for repetition of fundamental experiments. Purposeless reduction of experiment repetition will give rise to an engineering-educational lowering. So it is necessary to prepare some educationally effective arrangements, put it into practice of experiment and improve effectiveness of engineering education for students to learn from each cycle of experiment.

This paper describes an experiment example of Information systems engineering, which contains some sub-themes from hardware implementation to software production, our new trial to improve of effectiveness of engineering education and some kinds of experiment supporting environment.

2. Basic Concept for Themes of Experiments

Hardware and software seem to be inseparable elements of information (systems) engineering and work closely as if a tightly coupled pair of wheels. Knowledge and practical techniques about hardware and software are indispensable to information engineering. For the basic concept, therefore, any experiment related to information engineering has to include such a theme that concerns both hardware and software, while there may be another way to put a specific weight into either of them and to keep their balance among two or more themes of experiments.

Utilizing network is not only one of the educational subjects but also included in the essential techniques to upgrade our living space. It is no room for doubt that Web service on behalf of network is one of the most variable methods to distribute several kinds of teaching material. Network is also one of the most indispensable themes, which must be included in fundamental experiments. It can realize an efficient experiment for emphases on characteristics of information engineering to combine themes concerning hardware implementation and software production with utilization of network environment.

Conventional engineering experiments have been suffering from following problems:

1) because it is general to overemphasize actual experiences of making objects and routing, almost students take it for granted that every content of experiment can be realized easily and naturally,

2) a lot of students describe such reports that do not explain objective evaluation and analytic consideration but express only result of work and impression for routines during their period of experiment.

By means of instructing to analyze occurrence of errors qualitatively as well as quantitatively, elucidate cause of mistake generation and estimate ripple effects from malfunction, for example, it can be expected that reports with aimless impression decrease relatively and efficiency of experiment education will be elevated. Such a trial can lead to an agreeable appearance of experiment and change awareness for experiment from that achieving the targets of it seems natural to that students consider together why errors may occur.

In the former case of experiment, although a theme was given to a group of students, one or a few students undertook it and others used to be onlookers. On the other hand, in the latter case of experiment, every group needs additional roles to analyze occurrence of errors, elucidate cause of them and estimate ripple effects from them. The above change of awareness can provide some kind of tension for any group of experiment and improve educational effects of engineering experiment to build hardware and software.

An agreeable engineering experiment contains the following qualitative and quantitative analysis:

1) systematic classification of errors,

2) consideration of cause on error occurrence,

3) examination of frequency concerning error appearance, and

4) trial derivation of statistical properties on errors.

Such an analysis, therefore, can carry out an important role to discuss reliability of a whole system, introduce reliability engineering into engineering experiment with very few resistance and compose practical experiment through combination theme concerning hardware implementation and software production with some kinds of technique on reliability engineering

3. Configuration of Engineering Experiment

This section sequentially describes an overview of engineering experiment, details of hardware implementation and software production, and actual procedure of reliability analysis concerning to several types of errors on hardware and software.

3.1 Overview of Engineering Experiment

The title of our proposed engineering experiment is "Implementation of PIC-based Hardware and Production of Device Driver for Linux." This is one of the group-wise experiments to learn how to build a practical information system. As you know, this title explicitly specifies that the experiment contains subjects concerning both hardware and software. Additionally, subjects concerning on reliability analysis are implicitly included, too. The above experiment contains three sub themes for the following objectives:

Namely, 1) Understanding of electronic circuit and the digital circuit through actual hardware implementation, 2) Experience of systems programming to produce device driver software on Linux and Unix operations, and 3) Reliability engineering approach to classify some kinds of the mistake and evaluate quantitative error occurrences.

Figure 1 shows a general scheme and structure of our engineering experiment. After hardware is implemented, it is connected to Linux-based PC by means of parallel interface, where red line means "Centronics cable." Linux-based PC may play a role of server to control an implemented hardware such as external devices. Using in-house LAN of experimental room, several note PC's can be connected to Linux-based PC, and "telnet" and "ftp" commands are available to realize distributed computing environment among Linux-based PC and multiple note PC's

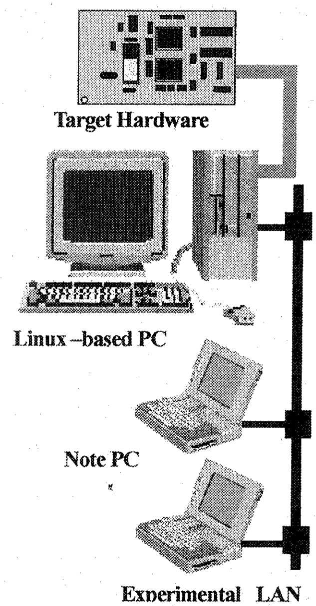

Figure 1 General View of Experiment

This experiment is prepared and managed for the third grade students of the department of reliability-based information systems engineering in Kagawa University. The framework of schedule for student's engineering experiment is illustrated in the Figure 2.

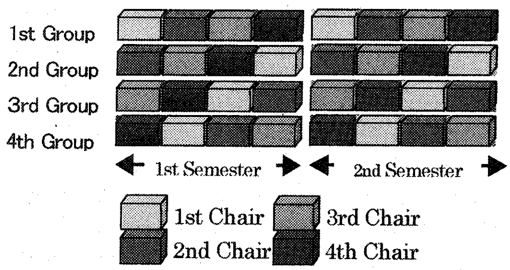

Figure 2 Schedule of Engineering Experiment

The department of reliability-based information systems engineering in Kagawa University has four chairs, and each chair prepares the according subject for its fundamental engineering experiment. Each chair must provide two types of experiments for 1st and 2nd semesters. Every experiment normally takes three or four weeks for twenty or twenty-four students to implement hardware, produce software and evaluate quantitative error occurrences. In our experiment four students are combined into one group and requested to work together and make a report about experiment cooperatively. The following three subsections explain the detail of experiment, such as implementing hardware, producing software and analyzing error occurrence.

3.2 Part One: Hardware Implementation

Hardware implementation part of experiment contains the three blocks of electronic circuits: electric power block, interface block and PIC microcomputer block. All hardware blocks are entirely implemented as electronic devices (i.e. IC's, LED's, etc.), capacitors, resistors, and switchers are bound on a print board by means of solder. Photo 1 shows a sample of the print board bound with several types of devices and elements.

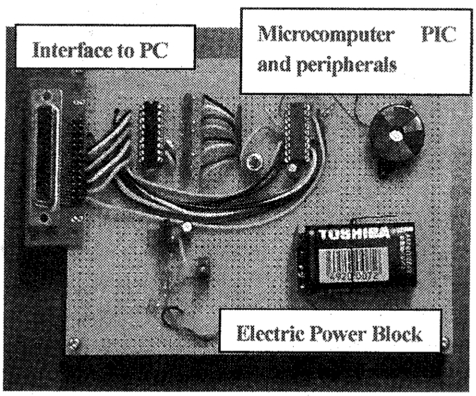

Photo 1 Sample of Implemented Print Board

Electric power block transduces from battery and generates regulated switching power for the whole system. Interface block facilitates connectivity between this hardware and PC through the 8-bits parallel cable. The eight numbers of LED's are prepared for indicating the bit size transferred from PC to the hardware and are convenient for checking whether hardware works correctly or not. PIC is an abbreviation of Peripheral Interface Controller. It can be customized by means of preloading control program into its internal EEPROM area. In this experiment, the control program preloaded in PIC plays a role to manipulate buzzer according to sequential signals sent from PC.

After devices and elements are bound on the print board, hardware verification must be performed by means of checking if every block can work suitably and all the devices and elements can operate correctly.

Photo 2 shows measuring instruments for hardware verification in our experiment. They are multi-function tester and oscilloscope. They are not so expensive.

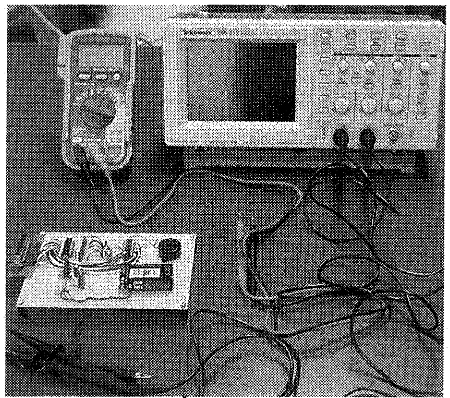
Photo 2 measuring instruments

A measurement can be achieved with the above instruments. And then hardware verification has been performed by measuring voltage, current or frequency for specific terminals. If either errors or incorrect operations are found during verification, debugging process must be beginning. And such errors and incorrect operations are target for reliability analysis, which will be described later.

3.3 Part Two: Software Production

After hardware level verification is over, total system verification phase will be possible to start. It has to be performed with connection between implemented (and verified) hardware and Linux-based PC. At the same time, this is the final stage of debugging for implemented hardware. Photo 3 shows that implemented hardware has been connected to Linux-based PC and pilot LED's of the hardware turn on.

The total system verification needs installing some kind of device driver on Linux to control implemented hardware from PC. Our experiment provides a sample of device driver for parallel port written in C program. Students can produce software of device driver for centronics parallel port by means of referring to the above sample one. A source program produced by students shall be compiled into object module by Gcc compiler on Linux. This compiling process can be performed through remote access from multiple note PC's to Linux server. In-house LAN facilitates a following distributed processing: 1) source files of device driver program are typed and edited in the multiple PC's environment, 2) they are got together into one source file on Linux server and compiled by remote terminal, and 3) generated object will be able to load on Linux kernel.

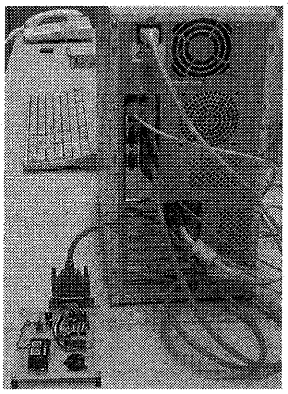

Photo 3 Connection of Hardware to Linux-PC

The software production theme of experiment contains some kind of UNIX literacy. After device driver program is correctly compiled into object file, the process of modifying Linux kernel must be started. This process requests students to play a role of "root (Administrator of UNIX)," who can operate Linux-based PC in the supervisor mode.

When device driver module is built into kernel, parallel interface is available for general (non-privileged) users to read from / write to the specific parallel port. The command "echo" is used to verify easily whether the specific port becomes available to be read/written or not. Photo 4 shows verification for implemented hardware is carried out through a series of UNIX commands.

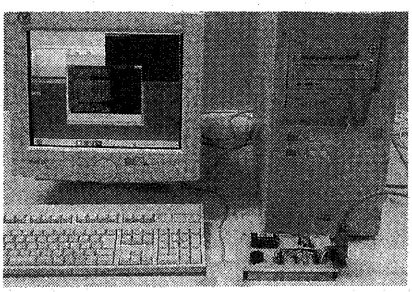

Photo 4 Verification of Implemented Hardware

In the software production process as well as hardware implementation process, several types of errors (including incorrect operations) are identified through the above verification. And such errors and incorrect operations are also handled by the reliability analysis, which will be described in the next subsection.

3.4 Part Three: Reliability Analysis

There have been several kinds of errors and mistakes from implementing hardware or producing software even in almost conventional experiments for student. In such cases, however, handling of errors and mistakes is seldom target of experimental themes. It may be natural that many types of conventional experiments have a special mission for students to obtain experience and skill to build systems regardless of any errors and mistakes. If effectiveness of experiment for students is not so essential, repetitive errors and mistakes at experiment will bring skill and ability of implementation and/or production into every student. This style spends a lot of time to educate students to become aware of reducing errors and mistakes in order to implement hardware and produce software effectively.

In our proposed experiment, both errors and mistakes are the very objects to be handled and analyzed. It will be very important and essential that some kinds of mechanism or framework are introduced to let students think about why errors occur and how mistakes are reduced. First of all, errors or mistakes must be classified or identified for each process of hardware implementation and software production. It is why some kind of qualitative analysis about errors and mistakes can be introduced into implementing/producing experiment. Table 1 explains a sample of errors classification.

Table 1 Sample of Error Classification

Error classification (Qualitative analysis)			
Hardware implementation		Software production	
Error id.	Position of occurrence	Error id.	Position of occurrence
Poor soldering	Power block, interface and PIC	Typing mistake	Source codes
Wrong polarity	Elements with polarities	Syntax error	Source codes
Ignorance for circuits	All over circuits	Errors from cooperation	Source codes
Human error	Anything else	Human error	Anything else

With classification of errors and mistakes, we can obtain several kinds of knowledge from error analysis. Moreover, such an analysis may give us fruitful suggestions to improve the process of hardware implementation and software production.

In order to take more advantage of error classification, objective analysis concerning errors and mistake should be additionally introduced, which is a second but not least point to handle possibly errors and mistakes in the quantitative procedure. In our experiment, a trivial style is employed so that students are requested to enumerate frequency of errors and/or mistakes in the classified categories, such as are classified in the Table 1. And students can obtain objective analysis by calculating ratio on error frequency as probability of error occurrence in the manner specified on Table 2.

Table 2 Sample of Error Evaluation

Error evaluation (Quantitative analysis)			
Hardware implementation		Software production	
Error id.	Probability	Error id.	Probability
Poor solder at Power block, interface and PIC	Number of poor solder / Number of all the solder	Typing mistake	Number of type mistaking / Number of all the statements
Wrong polarity	Number of bad polarity / Number of all the polarity	Syntax error	Number of syntax error / Number of all the statements
Ignorance for circuits	Existence of incorrect action from ignorance	Errors from cooperation	Existence of Errors from cooperation
Human error	More detail classification	Human error	More detail classification

4. Simple Evaluation and Perspective Problem

Information (systems) engineering education needs effective experiment, which can cover both fields concerning hardware and software. Ability of implementing hardware and producing software is essential for information engineers. Tightly coupled experiment with hardware issue and software one is desirable for engineering education.

We think that our experiment not only proposes a practical model of engineering experiment with hardware implementation and software production but provides an actual strategy to let students analyze why errors occur through the whole experiment and consider how to reduce errors. From some points of view, our experiment model may be more educational than other conventional experiments. For example, reports from students illustrate that they have paid attentions to analysis of error and reduction of mistakes. By means of their error classification, they have become aware that trivial errors like poor soldering and typing mistakes occur more often than they suppose at first. What is the most important key idea for students to classify several kinds of errors, analyze why errors occur and investigate how to reduce errors in the process of implementation and/or production? We do not know the very best solution, but the following style may be one of the candidates of solutions: 1) students make presentation about their errors which occur during implementation and/or production, 2) they also explain from what kind of viewpoints to classify their errors, 3) they prepare original countermeasures to reduce such errors, and 4) other group of students mutually evaluate that report and presentation with instructors.

There are a lot of problems in our experiment, especially problem on running cost is serious of all. Our experiment needs about 5,000 yen of material cost for each group. At the same time five or six groups begin to start their process of hardware implementation. All of our experiment requires less than 120, 000 yen for a semester. One of other problems is that instructors are necessary for hardware implementation and software production.

5. Concluding Remarks

A case study of engineering experiment is described. Its object is to provide an information engineering ability of implementing hardware and producing software. And it has a mechanism for students to pay attention to analyzing errors, which occur during implementation and production. Error classification and evaluation may make a time of implementing and producing more effective.

Acknowledgements

This study is partly supported by grand 12040107 in aid for scientific research from the Ministry of Education, Culture, Sports, Science and Technology.