Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Control Engineering Education with Experiments on Real-Time Control System Implementation

Byung Kook Kim

School of Electrical Engineering and Computer Science
KAIST (Korea Advanced Institute of Science and Technology)
Daejeon, Korea (email: bkkim@ee.kaist.ac.kr)

Abstract: Control engineering education at Electrical Engineering, Korea Advanced Institute of Science and Technology is discussed, which emphasizes experiments on real-time control system implementation. In addition to various lectures for theoretical control system analyses, appropriate hands-on experiments are performed for control system implementation utilizing up-to-date embedded system technology. Various useful experiments are constructed based on systems ranging from simple systems with microcontrollers to real-time systems with embedded Linux, in order to get firm understanding of control systems from theory to implementation.

1. INTRODUCTION

In the area of control systems education, theoretical analyses and syntheses are necessary. Mathematical backgrounds on control theories and computer-aided design methodologies are useful tools. In addition to those tools, hands-on experiments on real-time control system implementation are highly desirable to get firm understanding of control systems from theory to implementation.

In this paper, control engineering education at Electrical Engineering, Korea Advanced Institute of Science and Technology (EE KAIST) is discussed, which emphasizes experiments on real-time control system implementation. In addition to various lectures for theoretical analyses, hands-on experiments are performed for control system implementation utilizing up-to-date embedded system technology. Various useful experiments are constructed based on systems ranging from simple systems with microcontrollers to real-time systems with embedded Linux.

In Section 2, lectures on control engineering at EE KAIST are briefly discussed. Hands-on experiments with systems using simple low-cost microcontrollers are discussed in Section 3. In Section 4, a more advanced experiment based on embedded systems loaded with embedded Linux is described. We conclude this paper with concluding remarks in Section 5.

2. CONTROL ENGINEERING LECTURES

2.1 Undergraduate Lectures

Table 1 shows undergraduate courses on control engineering at EE KAIST. Brief description of each course follows.

EE202 Signals and Systems covers an introduction to continuous-time and discrete-time signals and systems, including Fourier series, Fourier transform, Laplace transform, and z-transform. Various types of systems with emphasis on linear time-invariant system are studied.

EE381 Control System Engineering covers general methods for analysis and design of dynamic systems. The main contents include modelling in the frequency and time domain, time response, reduction of multiple subsystem, stability, steady-state error, root locus technique, frequency response technique, and design via frequency response and state-space.

EE391 Electronic Control of Electric Machines discusses the operational principles, analysis, modelling and design of power conversion circuits in power electronics and carries out Spice simulations.

EE414 Embedded Systems discusses various hardware and software components and system implementation aspects of embedded system are covered. Covered topics include bus-based expandable ARM processor based board, open-source embedded Linux operating system, PC-based software development environment, digital and analog interface techniques, ARM assembly language, and device drivers. Hands-on experiments are added to enhance firm understanding.

Two major themes of EE481 Intelligent Systems are 'Modern Control System' and 'Computational Intelligence'. Each lecture will address a balanced emphasis on the theory about the control system and its applications in practice. The first part of this course includes digital control system design and state-space methods for control system design. Basic system identification scheme is also included, considering the control of unknown systems. Once background knowledge of the modern control system is established, this course will then focus on the second part composed of computational intelligence using fuzzy logic, artificial neural network, and evolutionary computation as main topics to introduce recent trend in intelligent control. Term projects will be assigned to test the algorithms to the given problems.

Table 1. Undergraduate Courses on Control Engineering, EE KAIST

Course ID	Subject Name	Credits
EE202	Signals and Systems	3
EE381	Control System Engineering	3
EE391	Electronic Control of Electric Machines	3
EE414	Embedded Systems	3
EE481	Intelligent Systems	3

2.2 Graduate Lectures

Table 2 shows graduate courses on control engineering at EE KAIST. Notable courses are described below.

The goal of EE505 Electronics Lab is to improve problem-solving techniques and ability for real design problems by performing electronic circuit designs and learning high-precision measurement techniques and error analysis methods. This course deals with five electronics experiments such as feedback amplifier design, controller design using microcontroller, digital signal processing using DSP boards, USB device driver programming, and RF experiments, each of which is conducted for two or three weeks.

EE581 Linear Systems include topics on system representation (input-output description, state variable description), solutions of linear dynamical equations, controllability and observability, irreducible realization, stability (BIBO stability, Lyapunov stability) for rigorous treatment of linear systems. In addition, feedback linearization is also covered.

EE582 Digital Control describes the analysis and design of digital control systems. Sampling and data reconstruction and Z-transform in computer control system will be covered. Analysis and design of digital control systems using frequency domain techniques is introduced. Also design of the digital control system using state space approaches is covered. As a term project, a real-time digital control system will be implemented on a microprocessor system.

EE584 Computer Aided Control System Design is intended for graduate students including senior level undergraduate students in engineering to introduce the concepts and techniques of the linear control system design. Students are encouraged to have a MATLAB experiment and asked to verify the computations using a MATLAB-SIMULINK software tool. Emphasis is placed upon application of the design technique to systems of interest to students.

EE594 Power Electronics Systems cover the design and analysis of the topology about the DC/DC converter, power factor correction circuit and control methods in that topology. Also topologies such as inverter, resonant converter, and active power filter are introduced, and the control algorithm of that topology is studied. Finally the state-of-the-art in power conversion system is discussed, and every student carries out a term project about design and modelling of power supply. On completion of this course students will have built confidence on their ability to design and analyses of the power conversion system.

EE681 Nonlinear Control is intended to present the fundamental result of analysis and design of nonlinear control systems. Especially, this course is concerned with the analysis tools for nonlinear dynamical systems and the design techniques for nonlinear control systems.

In EE682 Intelligent Control Theory, among the various well-known intelligent control techniques, the methods of fuzzy control and neural net based learning control are first introduced to allow for handling ambiguous / uncertain situations and effective supervised learning, respectively. Specifically, the theory of fuzzy sets and fuzzy logic-based inference mechanism are studied and the design techniques of fuzzy control are introduced. Then, the neural net learning structure is discussed, and the control system based on the artificial neural nets is studied. Fuzzy-neuro systems are also considered. In the second part of the course work, some other computational intelligence techniques such as GA (Genetic Algorithm) and the rough set are briefly covered and then the basic machine learning techniques and the reinforcement learning method are studied in conjunction with the their use in control system design.

EE683 Robot Control is intended to cover kinematics, dynamics and control algorithm of robot manipulator. After covering homogeneous transformations, kinematics equations,

Table 2. Graduate Courses on Control Engineering, EE KAIST

Course ID	Subject Name	Credits
EE505	Electronics Lab.	1:6:3
EE581	Linear Systems	3
EE582	Digital Control	3
EE584	Computer Aided Control System Design	3
EE594	Power Electronics Systems	3
EE681	Nonlinear Control	3
EE682	Intelligent Control Theory	3
EE683	Robot Control	3
EE686	Optimization Theory	3
EE687	Real-Time Control	3
EE783	Adaptive Control Theory	3
EE784	Supervisory Control Theory	3
EE785	Robust Control Theory	3
EE786	Optimal Control Theory	3
EE788	Robot Cognition and Planning	3
EE789	System Modeling and Identification	3
EE791	Power Conversion Circuits and Systems	3
EE792	Advanced Theory and Design of Electric Machines	3

motion trajectory planning, various robot control methods are handled, and the utilization of these control methods through simulation are studied.

EE686 Optimization Theory deals with optimization theories to solve problems in engineering, economics, management, and other practical applications. Classical methods based on geometry through linear vector space and function analysis are studied. Also optimization methods based on evolutionary computation and neural network are dealt with as an advanced technology. The contents include linear planning, nonlinear planning, dynamic planning, function optimization, least squares method, etc.

EE687 Real-Time Control is an important field in electrical engineering with applications to industrial automation, aerospace, and medical instrumentations. In this course, various topics for real-time control system are covered including performance, deadline, task scheduling, real-time operating system, real-time communication, and fault-tolerance.

More advanced control theories are handled in EE7xx courses.

3. STAND-ALONE CONTROL SYSTEM USING MICROCONTROLLERS

In this Section, experiments on control systems are discussed, which use microcontrollers and additional input/output circuit. Software for control application is designed and implemented.

3.1DC Motor Control System using Microcontroller

The main purpose of this experiment (a part of Electronics Lab) is to design and implement a DC motor velocity/position control system using a microcontroller. In two weeks, we give one hour lecture and a group with students perform 6 hour experiment or more per week with help from teaching assistants. Students can establish a firm concept on control system implementation, which is one of typical design examples with judicious combination of both hardware and software.

The problem is to design and implement a DC motor control system using a microcontroller. Given a DC motor to be velocity/position controlled, design and implement a suitable feedback controller using a microcontroller as a digital controller, H-bridge PWM amplifier as an actuator, and an optical encoder as a position sensor. Overall block diagram of this microcontroller-based DC motor control system is shown in Fig. 1.

Experimental steps include a sequence of development stages composed of getting acquainted with a microcontroller development system, actuating a DC motor with PWM (Pulse Width Modulation) switching amplifier, obtaining position feedback signal using optical encoder, and implementing a suitable feedback controller.

Microcontroller is a VLSI (Very Large Scale Integration) chip, which contains CPU (Central Processing Unit), memory, and input/output devices. A microcontroller named ATmega128 from Atmel (Atmel 2007) will be used, which has features of 8-bit RISC (Reduced Instruction Set Computer) CPU, program memory of 128 Kbytes, data memory of 4 Kbytes, and has many input/output interface devices. It is a typical small-but-powerful microcontroller. By integrating a suitable hardware and installing a suitable software program, one can implement a desired target system.

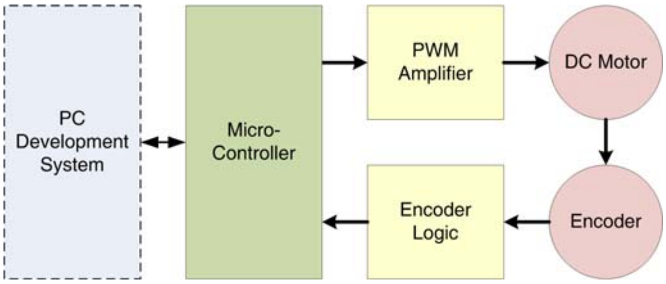

Fig. 1. Block diagram of DC motor control system using a microcontroller

In this experiment, for rapid and convenient implementation, a PCB board including ATmega128 will be utilized. The ATmega128BK3 board from Interboard Co. Ltd (Interboard 2005) is selected, which is shown in Fig. 2. This board includes an ATmega128 microcontroller, SRAM (Static Random Access Memory) of 32 Kbytes, a RS-232C driver, a 16 MHz crystal oscillator, a reset switch, and also an ISP (In-System Programming) connector.

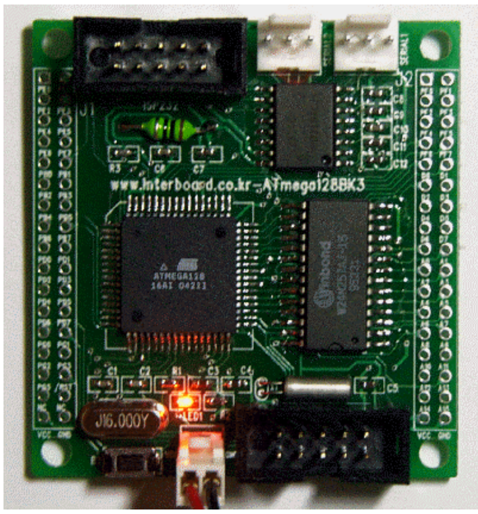

Fig. 2. Atmega128BK3 Microcontroller Board

Cross development system is a software development system for developing software which is to be executed in the target embedded board. As a suitable hardware, a general-purpose computer is necessary including a keyboard and display for user input/output, and also a disk for program storage. Since vast amount of PCs (Personal Computers) are widely available, a PC can be utilized as a development system (sometime called a host computer or a host development system). Suitable cross-development development software should be installed and utilized, which includes an editor, a compiler, a linker, and a debugger. The compiler here does not generate Pentium binary codes, but generates Atmel binary codes. Hence this compiler is called a cross compiler, and the system is called a cross-development system. Software for development system is often called development tools.

A C compiler performs a conversion function from a C language program to a machine language program (or an object code). The AVRGCC has good performance, which is freely downloadable from Gnu (AVRGCC 2005). In the AVRGCC package, a linker is also included, which combines multiple object codes and suitable libraries, and then generates an executable code. We use the AvrEdit package, which contains the AVRGCC as well as a graphic user interface (AvrEdit 2007).

The ATmega128BK3 board supports ISP (In-System Programming), which is a function to write the executable code into the program memory (or flash memory). A representative program, the PonyProg2000 with version of V2.06C (PonyProg 2007), is used.

The plant to be controlled in this experiment is a DC motor. A general-purpose DC motor with gear will be used, and an external encoder for position sensing will be designed and attached to the DC motor.

A variety of DC motor requires driving voltage from 3 to 400 V, and driving current from 1 to 50 amps, and hence 5V logic voltage is insufficient to drive them: A bi-directional power-driver as well as a suitable power-supply is required. An H-bridge type switching power amplifier will be used, since it requires a single power supply only. In this experiment, a commercialized H-bridge IC will be used.

Principle for the optical encoder is to detect the disk rotation via slotted disk and utilizing a light transmitter and a light sensor. In order to detect the rotational direction also, two sets of light transmitter and sensor are utilized to get 90 degree phase difference signals A and B. Also a marker pulse, usually denoted as Z, may be obtained to detect a certain position in the disk rotation in some encoders. By detecting both rising and falling edges in both signals A and B, one can obtain the resolution improvement of 4 times.

A microcontroller functions as a digital controller. The reference input is obtained from the development PC via RS-232C serial port. Here, the unit of degree is used. The current position is sensed via encoder logic, which is fed by encoder signals A and B. The position error is computed, and a suitable discrete-time PID controller algorithm is programmed to compute the controller output.

The value of PID output is converted into a PWM signal and transferred to the PWM generator. These sequences should be repeated for each sampling interval T, which is equal to 5 ms. The sampling interval T can be generated by the timer 0 in ATmega128.

This is a two weeks experiment. In week 1, students design hardware circuits and its test programs.

A PWM amplifier and encoder detection logic circuit is designed using the provided components. For the PWM generation, the section on counter in the datasheet of ATmaga128 should be understood, and the method fro driving the counters should be extracted.

For the actuation of the DC motor, the counter 2 in the ATmega128 is used to generate a PWM signal. The PWM signal is transferred to the H-bridge switching amplifier.

Encoder detection logic may be devised as follows. Use counters 1 and 3 in the ATmega128. Counter 1 can be used as an up-counter, and counter 3 can be used as a down counter. Thus, all the four counters in ATmega128 are fully utilized. Using a program, the difference of up and down counters can be calculated, which results in the position of the motor. Using this method, a separate up/down counter is not necessary, and can simplify the overall circuit.

Students design and write a PWM test program which constantly generates 10 kHz PWM wave with a certain duty cycle. They also design and write a PWM and encoder test program which gets two inputs from up and down counters and accumulates. Transmit the count output to RS-232C every 1 sec. The result can be seen in the PC display via Hyperterminal program.

In week 2, students design and implement a PI (Proportional Integral) velocity/position controller programs using the C language. The values of proportional gain and the value of reference input are assumed fixed inside the program.

3.2 Mobile Robot System using Microcontroller

The purpose of this experiment is to design and implement a mobile robot using a microcontroller and two DC motors in two weeks – one hour lecture and 6 hour experiment per week.

The problem is to implement and drive a Mobile Robot using a microcontroller. Given a DC motor to be velocity controlled, design and implement a suitable feedback controller using a microcontroller as a digital controller. Complete simple circuit and generate a velocity profile for driving a mobile robot. Overall block diagram is shown in Fig. 3.

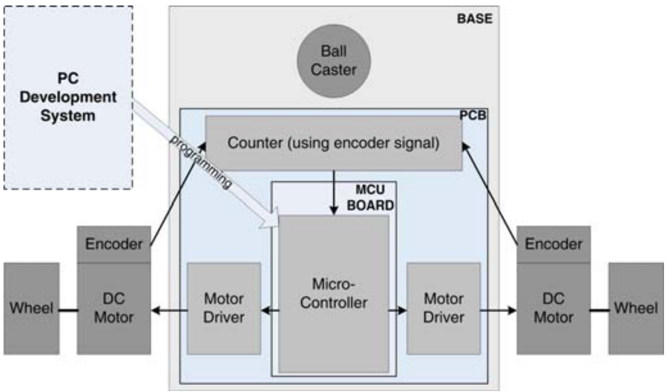

Fig. 3. Mobile Robot using a Microcontroller and two DC Motors

Two wheels and a caster are used to actuate the mobile robot. A suitable frame is constructed using a scientific construction kit. Two DC geared motors with encoders are used to drive wheels.

A microcontroller board ATmega128BK3 functions as the main controller. Reference input is given from the user via a serial port in the development PC. Then a suitable trapezoidal velocity profile with maximum velocity and acceleration is generated by a suitable program by students.

Two encoder signals from DC motors are read via separate encoder counter logic which is programmed in a CPLD. A suitable application program is designed and tested to perform velocity control of two DC motors using PWM generators and motor drivers (H-bridge power amplifiers). This procedure is repeated at each sampling time.

Students' opinion on these courses: They had very good opportunities to experience system concept of microprocessor-based digital control system implementation as well as various component techniques for sensors and actuators.

4. EMBEDDED CONTROL SYSTEM WITH OPERATING SYSTEMS

In this Section, an experiment on embedded control system with operating system is described.

4.1 Voice Recorder using Embedded Linux

Students design and implement a voice recorder within 6 weeks (one hour lecture and 6 hour experiment per week). This Lab is for senior undergraduate students. We adopt a representative microprocessor embedded system board EZ-X5 (Falinux 2007), in which an Intel PXA255 Xscale processor with ARM core (Intel 2007) is the central processing unit as shown in Fig. 4. An open-source operating system embedded Linux is installed in the flash memory. Various analog and digital input/output interfacing experiments are performed and a voice recorder is implemented, to get a strong basis for embedded system design.

Major specifications of EZ-X5 board are as follows:

- PCB 100 mm x 140 mm
- MCU 400MHz PXA255 ARM RISC Chip ARM10
- RAM 64 Mbytes SDRAM
- ROM1 512Kbytes Boot Flash
- ROM2 64Mbyte NAND Flash
- Ethernet CS 8900, 10-Mbps
- Serial RS – 232C 3Ports
- USB USB Client
- LCD 640x480 TTL-Port
- Touch 4 wire type
- LED Debugging 4 Bits
- JTAG ON – Board JTAG Converter
- Extension Connector 160-pins Board to board Connector

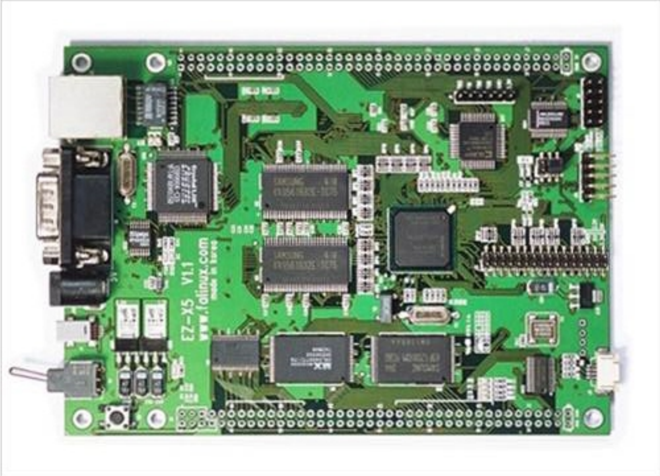

Fig. 4. EZ-X5 Embedded System Board

Six topics are covered. In Exp. 1, Linux software development environment is constructed in a PC. A display interface with 7-segment LEDs (Light Emitting Diodes) and discrete LED lamps are constructed in Exp. 2. A suitable display driver module is also studied. In Exp. 3, a keyboard interface is constructed to implement a simple man-machine interface. In Exp. 4, an audio playback circuit is implemented using an audio codec chip. In Exp. 5, functionality of audio playback is constructed. In the Final Exp. 6, an ADPCM (Adaptive Delta Pulse Code Modulation) algorithm is constructed, and the overall voice recorder is implemented.

During the six Experiments, a suitable interface board composed of display and keyboard logic as well as audio codec is to be implemented as shown in Fig. 5.

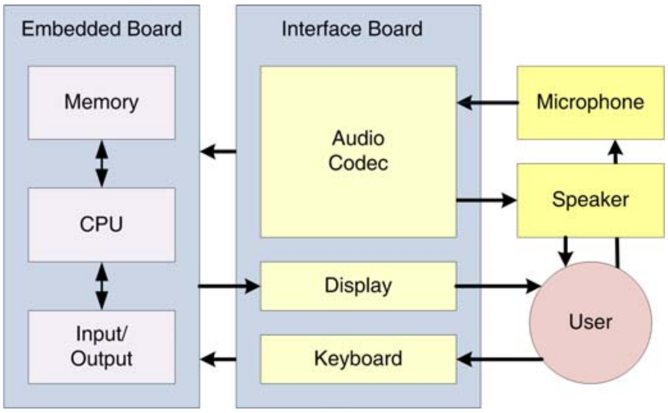

Fig. 5. Block Diagram of a Voice Recorder Implementation

In Exp. 1, students get acquainted with the cross development system:

- Target board: EZ-X5 board with EzBoot and Embedded Linux
- Host computer: PC with Linux, cross compiler, NFS, and minicom

Software development systems are constructed as shown in Fig. 6. Embedded Linux software composed of a kernel and ramdisk is installed in the flash in the EZ-X5 board. A Linux operating system and a cross-compiler software from Gnu are installed in the development PC. The minicom software is utilized to send and receive signals from a serial port of EZ-X5 board, which functions as a console for the embedded Linux.

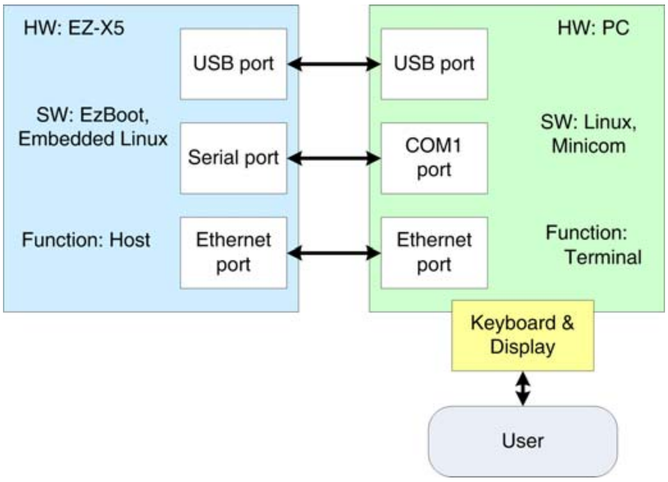

Fig. 6. Cross Development Environment for Systems with Embedded Linux

In Exps. 2 & 3, keyboard and display interface as shown in Fig. 7 is constructed. Drivers for keyboard and display should be built as modules in the kernel level, in order to get direct access to hardware as well as getting faster response.

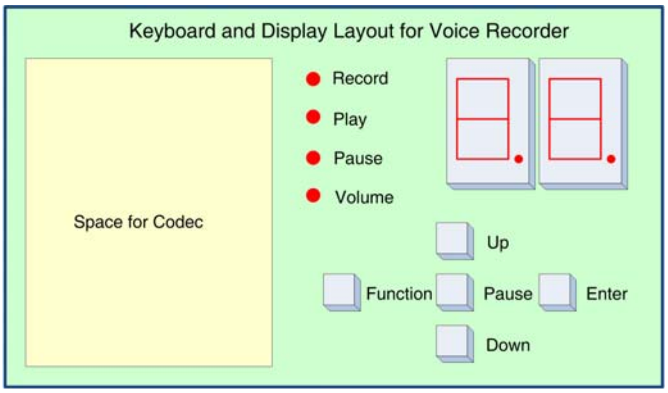

Fig. 7. Keyboard and display interface of voice recorder.

In Exps. 4 & 5, we adopt a codec chip with AC97 interface: CS4202 from Cirrus Logic, which is AC'97 2.2 compliant, integrated microphone preamplifier (Cirrus Logic 2007). Since the Xscale processor has an AC97 interface, connection of CS4202 chip and Xscale chip is quite easy.

In Exp 6, all the hardware constructed so far is integrated into one board, and complete voice recorder software along with ADPCM compression/decompression is designed and implemented.

In this experiment, we still are not using a real-time operating system. However, it is quite straightforward to adopt a real-time operating system such as RTAI (Real-Time Application Interface) (RTAI 2007) into the flash memory of EZ-X5 board.

5. CONCLUDING REMARKS

Hands-on experiments on real-time control system implementation are highly desirable to get firm understanding of control systems from theory to implementation. Control education at EE KAIST is discussed, which emphasizes experiments on real-time control system implementation. In addition to various lectures for theoretical analyses, hands-on experiments are performed for control system implementation utilizing up-to-date embedded system technology. Various experiments are constructed based on systems ranging from simple microcontrollers to real-time systems with embedded Linux.

Control systems with simple microcontrollers are quite useful for control education, with low-cost.

After completing control systems with embedded Linux, students can build more complex embedded control systems as thesis or research works. Some examples are as follows:

1. Mobile robot design and implementation using embedded Linux can be performed for a thesis work. Various sensors along with suitable software can be added for environment sensing and learning.

2. Distributed real-time control of robots: More advanced intelligent service robot can be built based on multiple processors connected with a suitable network. Various tasks are distributed into processors to get the best performance possible.

REFERENCES

Atmel (2007). *"ATmega128 User's Manual"*, http://www.atmel.com.
AvrEdit (2007). *"AvrEdit3.7c"*, http://www.terabank.co.kr.
AVRGCC (2005), http://www.avrfreaks.net/AVRGCC.
Cirrus Logic (2007). *"CS4202 Data Sheet"*, http://www.cirruslogic.com.
Falinux (2007). *"EZ-X5 Embedded Board"*, http://www.falinux.com.
Intel (2007). *"PXA255 User's Manual"*, http://developer.intel.com.
Interboard (2005). *"ATmega128BK3 Board"*, http://www.interboard.co.kr.
PonyProg (2007). *"PonyProg 2.06C"*, http://www.lancos.com/prog.html.
RTAI (2007). "RTAI", http://www.rtai.org.