논문 2009-1-9

웹 기반 감시제어시스템을 위한 임베디드 시스템 기술

A Embedded System Technology for Web based monitoring and control system

박종진*, 최규석**

Jong-jin Park, Gyoo-Seok Choi

요 약 본 논문은 웹 기반의 감시제어시스템을 위한 임베디드 시스템 기술의 구현 예를 제시하였다. 웹 기반의 감시제어시스템을 위해 임베디드 시스템인 타겟 보드와의 인터페이스를 개발하고 이를 특정 임베디드 시스템인 온실 모델에 적용하여 그 실용성을 확인하였다. 자바 클래스를 이용한 클라이언트/서버 소켓 프로그램을 구현하였고, 자바 애플릿 기반의 사용자 인터페이스(GUI)에 의한 웹 기반 온실 감시제어시스템을 구축하였다. 구축된 웹 기반 감시제어시스템은 TCP/IP 상에서 온실의 정보를 클라이언트 프로그램에 잘 전달하여 표시하며 웹 상에서 동작하는 자바 애플릿 클라이언트에서 보내는 제어 신호를 서버를 통해 온실 모델에 잘 전달하여 동작시키는 것을 볼 수 있었다.

Abstract In this paper, an example of implementation of a embedded system technology for web based monitoring and control system is presented. For it interfaces with target board as an embedded system was developed, which was applied to specific green room model and verified of its usefulness. We implemented client/server socket programs using Java classes, and web based green room monitoring and control system as an user interface using JavaApplet. The implemented system did send information of green room model to client programs well on TCP/IP and control signals from client to green room model well too.

Key Words : 웹 기반 감시제어시스템, 임베디드 시스템, 자바 애플릿, 온실 감시제어시스템

I. 서 론

인터넷은 전국적 국가 기간망으로 성장하여 초고속 인터넷이 가정과 직장 등 거의 모든 곳에 연결되고 있다. 인터넷을 이용하면 인터넷이 연결된 곳에서는 어느 곳에서나 웹 브라우저나 GUI 응용 프로그램을 통해 접근 가능하므로 다양한 서비스나 시스템을 구현할 수 있다. 이러한 인터넷을 이용한 원격 감시제어기술이 개발되어 다양한 시스템이 개발되고 서비스가 제공되고 있다.

최근에는 인터넷을 이용한 웹 기반 원격 감시제어시스템 기술과 임베디드 시스템 기술이 결합되어 다양한 제품이 개발되고 있다. 임베디드 시스템은 응용분야에 따라 다양한 설계 및 구현 상의 제약 조건들을 내포하고 있다. 따라서 임베디드 소프트웨어 개발 과정은 이러한 제약 조건을 충실히 반영하여야 한다. 그러나 다양한 제약조건으로 인해 임베디드 시스템을 위한 소프트웨어를 개발하기 위해 일반화된 소프트웨어 공학 이론에 근거한 개발 방법은 의미가 없으며 응용분야에 최적화된 개발 방법론을 제시하는 것이 우선된다[1-5].

본 연구에서는 인터넷을 통한 웹 기반 감시제어시스템의 개발을 위해 프로토타이핑 도구로서 레고 사의 레고 블록을 이용한다. 레고 블록은 쉽게 조립할 수 있고 재사용 가능한 부품 및 블록들을 활용하여 실제 개발하고자 하는 실물과 유사한 모양과 기능을 가진 실물 프로토타입을 개발하여 소프트웨어를 개발함으로써 웹 기반

*정회원, 청운대학교 인터넷학과
**종신회원, 청운대학교 컴퓨터학과 (교신저자)
접수일자 2008.12.10, 수정완료 2009.2.3

임베디드 감시제어시스템을 효율적으로 개발할 수 있다. 이를 위해 시스템에 내장되는 타겟 보드와의 인터페이스를 구축하고 개발된 시스템을 레고 블록을 이용한 온실 모델의 온도제어에 적용하여 그 동작을 확인하고 그 실용성을 입증하였다. 실제 온실 감시제어에 사용되는 소프트웨어는 클라이언트-서버 방식의 인터넷 프로토콜을 이용하여, java applet으로 클라이언트(client)를 작성하고 응용 프로그램에 네트웍 서버를 추가하여, 둘 사이에 TCP/IP 프로토콜로 통신하는 방식을 사용하였다.[6-9]

II. 시스템 설계 방법

1. 임베디드 보드

인터넷 기반 원격 감시제어시스템을 구현하기위해 자원을 엑세스하여 사용될 레고 부품에 맞춰 그들을 이용할 수 있는 인터페이스 센서 보드와 액츄에이터 보드를 제작하였다. Sensor 접속 보드는 RCX의 Active sensor 구동 방법을 본 떠서 만든 것으로 원래 RCX는 10V의 전압을 이용해 3mS동안 8V를 공급하다 0.1mS동안 5V를 공급하며 이 5V 구간에서 sensor의 출력 값이 나타나게 되어 있다. Sensor 보드는 7.5V의 전압을 약 0.7mS동안 공급하다 0.1mS동안 5V를 공급하며 이 5V 구간에서 출력 값이 나타도록 구현되어 있다. 다음 그림 1은 구현된 센서 보드의 회로도의 일부를 보여준다.

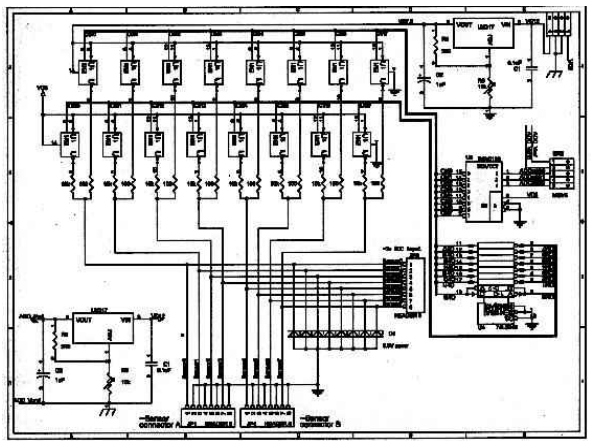

그림 1. 센서 보드의 회로도
Fig 1. Circuit diagram of sensor board

액츄에이터 보드는 레고 블록에 내장된 모터를 구동하기위한 보드로서 5V, 12V전원으로 4개의 L298 chip을 이용해 총 8개의 모터를 구동할 수 있게 하였으며 메인 보드에서 출력되는 PWM signal을 받아서(input) 2개의 header에 각 4개씩 모터에 output단자를 연결하도록 되어있다. 다음 그림 2는 구현된 액츄에이터 보드의 회로도의 일부를 보여준다.

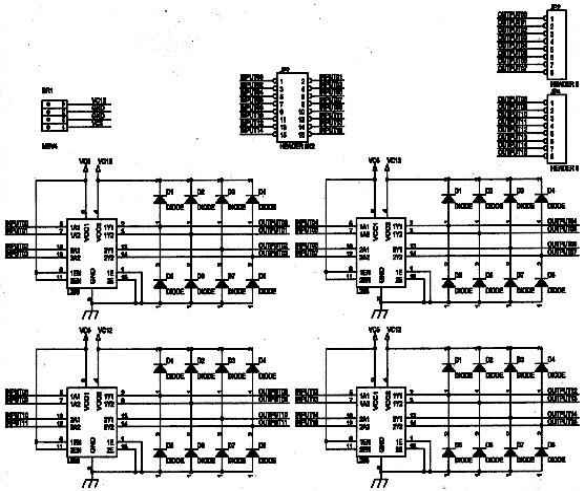

그림 2. 액츄에이터 보드의 회로도
Fig 2. Circuit diagram of actuator board

2. 웹 기반 감시제어시스템 소프트웨어

개발된 인터페이스 보드를 통한 웹 기반 감시제어시스템의 소프트웨어 개발은 Java와 Java Applet을 이용하여 클라이언트/서버 모델에 기초하여 작성하였다. 자바의 소켓 통신을 이용하여 원격감시제어에 필요한 프로그램 작성방법을 사용하였는데 네트워크에서 데이터가 전송되기 위해서는 패킷(Packet)이라는 단위로 쪼개져서 데이터가 전송되고, 상대측에서 이러한 패킷을 합쳐서 데이터를 복원한다. 이러한 동작을 해주는 것이 바로 TCP/IP 프로토콜이다. 소켓은 TCP/IP 네트워크 통로의 끝처럼 생각할 수 있는데, 현재 소켓은 TCP/IP 프로토콜을 사용하는 네트워크 프로그램을 작성하는 표준으로 사용되고 있다. 자바에서 소켓을 이용하여 데이터를 전송하기 위해서는 Java API의 java.net.Socket 클래스를 이용한다.

클라이언트/서버 모델에서 소켓 프로그래밍을 하기 위해서는 클라이언트 역할을 할 때와 서버 역할을 할 때 Socket 클래스를 사용하는 방법에 차이가 있기 때문에 프로세스가 서버 역할을 할 것인지, 클라이언트 역할을 할 것인지를 먼저 결정한다. 소켓 클래스를 클라이언트에서 사용하기 위해서는 아래의 그림과 같은 3단계를 거

친다.

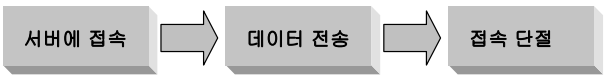

그림 3. 클라이언트에서의 소켓 사용법
Fig 3. Socket usage in Client

1) 서버에 접속

Socket 클래스를 이용하여, 서버에 접속하기 위해서 호스트의 주소와 포트 번호를 소켓 클래스의 생성자에 넘겨줌으로써 원격호스트의 서버 프로세스에 접속하게 된다.

2) 데이터 전송

일단 Socket 객체가 생성되면, 입출력 스트림과 연결할 수 있다. 소켓 클래스 객체의 getInputStream(), getOutputStream()메소드를 이용하면, InputStream과 OutputStream을 구할 수 있다. 일단 스트림을 구하면, BufferedReader등의 필터 스트림을 사용하면 편리하게 프로그래밍 할 수 있다. 이것은 네트워크의 특성상 데이터의 전송 속도가 일반 입출력에 비해서 느리고, 데이터가 중간에 손실되는 경우도 많기 때문이다.

3) 접속단절

데이터의 전송이 완료되었으면, 원격호스트에 접속을 단절한다.

서버는 클라이언트가 세션을 요청하면, 이에 대한 응답을 하는 프로세스로서 서버도 클라이언트와 마찬가지로 소켓 클래스를 이용한다. 서버에서는 클라이언트와 달리 상대측의 접속을 대기하고 있어야 한다. 자바에서는 이러한 기능을 가진 클래스가 제공되는데, 이를 ServerSocket 클래스라고 한다. 아래 그림 4에서처럼 ServerSocket 클래스는 특정 포트에서 대기하고 있다가, 접속 요청이 있으면 이를 새로운 소켓으로 연결하여 처리해준다.

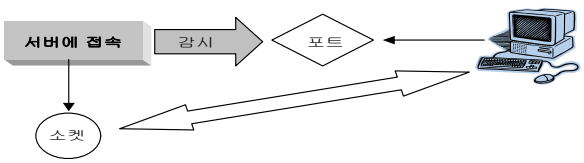

그림 4. 서버 소켓의 구조
Fig 4. Structure of Server socket

보통 서버에는 하나 이상의 클라이언트가 접속하기 때문에, 서버는 여러 개의 접속을 처리할 수 있도록 하기 위해 쓰레드를 이용하여 작성하였다. 서버의 소켓 클래스 사용은 다음과 같다.

1) ServerSocket 클래스의 생성

서버 소켓을 이용하기 위해서 ServerSocket 클래스를 이용하여 객체를 생성한다. ServerSocket 클래스객체는 생성 시에 감시할 포트 번호를 넘겨준다.

2) ServerSocket 클래스의 접속 처리

서버 소켓 자체는 클라이언트에서 접속 요구가 들어왔는지 감시하는 역할만을 담당할 뿐 데이터를 읽거나 쓸 수 없다. 따라서 클라이언트에서 일단 접속 요구가 있으면, 이를 소켓 객체에 다 할당하고, 할당된 소켓 객체가 실제 접속을 담당하게 되는 구조를 갖는다. 이를 위해서, 서버 소켓의 accept()메소드가 사용된다. accept()메소드는 호출된 뒤, 접속 요청이 들어올 때까지 지연하고 있다가, 요청이 들어오면, 요구한 클라이언트 측과 접속을 설정한 후, 해당 접속을 유지하는 Socket 클래스 객체를 넘겨준다.

3) ServerSocket을 이용한 다중접속 서버 프로그램

한번에 하나의 클라이언트만을 지원하는 것이 아니라 동시에 여러 클라이언트의 접속 요구를 지원할 수 있도록 하는 다중 접속 서버(Concurrent Server)를 만들기 위해서 서버 소켓과 accept()메서드에 의해 생성된 소켓간의 긴밀한 협조가 필요하다. 다중 접속 서버를 만들기 위해서 서버 소켓이 accept()메서드를 얻을 때, 해당 소켓을 쓰레드로 처리해주고, 서버 소켓은 계속해서 접속 감시만을 담당하였다.

III. 감시제어시스템 설계 및 결과

개발된 임베디드 인터페이스 보드와 소프트웨어를 특정 임베디드 시스템인 온실 모델에 적용하여 그 실용성을 확인하였다. 현재 국내에 보급되어 있는 작물재배를 위한 온실 감시제어시스템은 노동 집약적 시설에 각 장치를 제어하는 on/off 식 제어반으로 구성되어 있는 경우가 많다. 이러한 시스템은 하드웨어적인 결함, 야간이나 원격지에 있을 경우에는 오류 발생 등에 대한 온실의 상

태파악이 어렵다. 따라서, 기존의 on/off 식 제어 방식을 인터넷 상에서 원격지의 데이터 획득과 온실의 상황을 모니터링할 수 있는 서버-클라이언트 시스템의 설계를 통해 대체함으로써 온실 관리시스템의 효율성을 증가시킬 수 있다.

1. 웹 기반 온실 감시제어시스템

TCP/IP 프로토콜에 기반한 Java 소켓 프로그래밍을 이용하여 인터넷 기반의 온실 감시제어시스템을 구축하기 위해 클라이언트의 접속을 대기하는 서버(Server) 모듈과 온실 시스템을 제어하기위해서 접근하는 클라이언트(client) 모듈을 구현하였다. 온실 감시제어시스템의 설계를 위해 레고 블록을 이용하여 온실의 모델을 만들고 이를 인터페이스 보드를 이용하여 컴퓨터 시스템과 연결하였다. 실제 온실 시스템이 갖추어야할 기능은 다음과 같이 생각할 수 있다.

- 온실 내의 온도를 측정
- 온실 내의 광도를 측정
- 온실 내의 온도가 올라가면 식히기 위해 창을 연다.
- 온실 내의 온도를 빨리 식히기 위해 팬을 동작시킨다.
- 온실 내의 광도가 떨어지면 램프를 동작 시켜 광도를 유지

실제 온실 시스템이라면 위와 같은 기능들이 있어야 하겠지만, 우리는 그 중에서 광도 측정과 광도 유지용 램프 부분을 제외시키고 기타 부분에 대해서 레고 시스템을 구성하였다. 다음은 모델에서 사용될 기능들이 어떻게 작동되는 지를 나타내는 시나리오이다.

- 온실 내부의 온도가 지정한 온도 이상으로 올라가면 온도센서가 이를 감지한다.
- 내부의 온도를 내리기 위해 온실의 창문을 연다.
- 창문을 연 후, 온실 의 온도를 내리기 위해 팬을 동작시킨다.
- 온도센서가 감지한 온도가 지정온도이하로 내려가면 팬의 동작을 멈춘다.
- 다시 창문을 닫기 위한 창문 제어 모터가 동작하고 이 기간 동안 온실내의 광도를 보충해주기 위한 램프도 동시에 동작한다.
- 창문이 닫히고 램프가 꺼지면서 지정된 작업을 마무리한다.

그림 5는 인터넷 기반 온실 제어 시스템의 전체 구성도 나타내며 클라이언트에서 명령을 보내면 네트워크(Network) 망을 통하여 서버 측에 전달되고 서버는 클라이언트에서 보내준 명령을 처리하고 이에 따라 온실을 제어 한다.

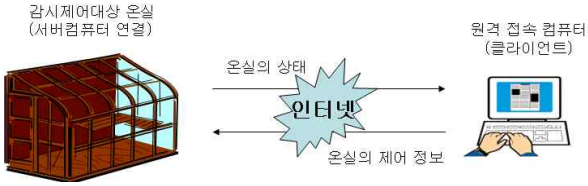

그림 5. 전체 구성도
Fig 5. Overall structure of the system

그림 6은 레고 블럭을 이용하여 구현된 간략한 온실의 모형도이다.

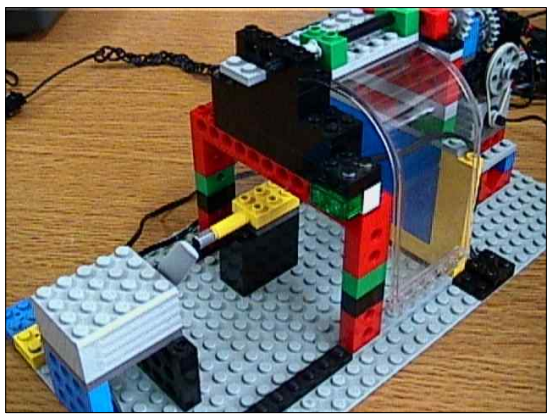

그림 6. 레고 블록 온실 모형도
Fig 6. Green room model using Lego blocks

2. 클라이언트/서버 모듈 프로그램

인터넷 기반 온실 제어시스템 구현을 위해 Java에서 제공하는 Socket API를 이용하여 클라이언트와 서버용 모듈을 작성하였다. 그림 7은 웹 기반 온실제어를 위한 전체적인 프로그램 도식도이다.

가. 클라이언트 프로그램 구현

클라이언트 프로그램의 기능은 크게 둘로 나눌 수 있다. 사용자와 대화하는 사용자 인터페이스와 서버와의 통신을 담당하는 소켓프로그램으로 나눌 수 있다.

1) 사용자 인터페이스 프로그램
(GreenHouseApplet.java)

이 코드는 웹 브라우저에서 실행하기 위해서 java Applet를 사용하여 만들었으며 위의 신호표를 참고하여 명령어 버튼들을 만들었다. 이것은 웹 브라우저의 플러그-인(Plug-in) 형태로 실행이 되며, 초기화시 사용자 인터페이스를 배치하고, 서버와의 통신을 위해 Connect객체를 생성 하며, 사용자와의 대화를 위하여 이벤트 처리 루틴을 만들었다.

2) 클라이언트 측 소켓 프로그램(Connect.java)

서버와의 소켓 통신을 위하여 필요하며 주요 기능은 사용자에 의해서 발생된 시그널을 서버 측에 보내고, 서버 측으로부터 보내어지는 온도센서, 온실창의 개폐 상태대한 값을 받아들인다. 이때 서버와 문자 데이터로 통신하지 않고 객체로 통신 한다. 즉, 클라이언트의 제어 명령을 객체에 담아서 보낸다. 소켓 프로그램에서 객체를 사용하여 통신 할 경우 주의해야 될 점은 보낼 데이터가 있을 때마다 새로이 생성하여 보내야 한다는 것이다.

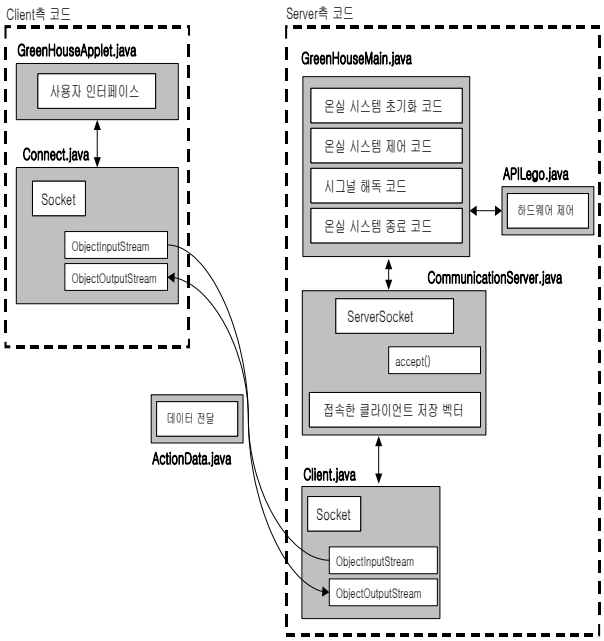

그림 7. 전체적인 프로그램 도식도
Fig 7. Overall diagram of programs

나. 서버 프로그램 구현

server의 기능은 크게 네 부분으로 나눌 수 있다.

① 클라이언트의 접속을 감시 및 클라이언트 접속 관리 부분

② 클라이언트와의 통신을 위한 부분.

③ 온실 시스템을 제어 및 전체 시스템을 관리하는 부분.

④ 하드웨어를 제어 하는 부분.

원격지의 사용자와 통신을 위한 부분에 대한 프로그램은 다음과 같다.

1) 클라이언트 접속 감시 및 관리 프로그램 (CommunicationServer.java)

클라이언트의 접속을 감시하는 코드는 서버에 특정 포트로 접속하는 클라이언트의 접속을 대기 하고 있다가 클라이언트의 접속이 이루어지고 나면 vector의 저장되어 있는 Client객체를 하나씩 읽어서 서버 측에 정보를 보낸다. Client의 객체는 서버에 접속하고 있는 사용자 수만큼 가지고 있다. 반드시 접속자 수와 vector의 크기는 일치해야 한다.

2) 클라이언트와의 통신 프로그램(client.java)

클라이언트와의 통신을 담당하는 코드로 앞에서 설명한 Connect.java 파일과 같은 구조를 가진다. 이 코드에서도 객체 통신을 위하여 ObjectInputStream, ObjectOutputStream 객체를 사용한다. 만약 서버 측에서 다른 형태의 데이터 타입으로 보낸다면 클라이언트의 서버 측에서 보내준 데이터를 받는 부분도 바뀌어야 된다. 즉 서버와 클라이언트간의 통신을 같은 데이터 포맷으로 주고받아야 된다.

3) 서버 실행하기

웹 서버로는 Apache 웹 서버를 사용하였다. 웹 서버를 실행시킨 후 서버 프로그램(GreenHouseMain.java)를 실행한다.

4) client 실행하기

클라이언트를 실행하기 위해서 앞에서 만든 자바 애플릿(GreenHouseApplet.java)을 포함하는 html파일을 만들어 실행한다. 다음 그림은 웹 브라우저를 통하여 서버에 접속한 후 온실감시제어 프로그램을 실행한 결과 화면이다.

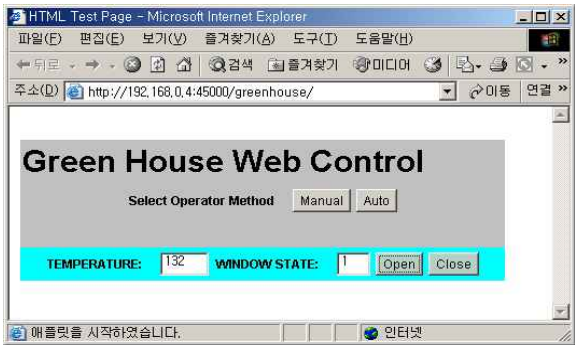

그림 8. 클라이언트 실행 결과 화면
Fig 8. Result of web-page in Client

IV. 결 론

본 논문은 웹 기반의 감시제어시스템을 구현하기 위해 임베디드 시스템 기술을 이용하여 레고 블록으로 이루어진 타겟 보드와의 인터페이스를 구현하고 인터넷 프로토콜인 TCP/IP 상에서 Java 기술을 활용한 웹 기반의 감시제어시스템 설계에 대한 예를 제시하였다. 이를 위해 Java의 Socket 클래스를 이용한 클라이언트/서버 소켓 프로그램을 구현하였고 이를 이용하여 온실의 온도 감시제어시스템을 구축하였다. 온실의 모델은 레고 블록을 이용하여 만들고 이를 개발된 임베디드 인터페이스 보드와 연결하고 클라이언트의 접속을 대기하는 서버(Server) 모듈과 온실 시스템을 제어하기위해서 접근하는 클라이언트(client) 모듈을 구현하였다. 구축된 인터넷 기반 온실 감시제어시스템은 온실의 정보를 클라이언트 프로그램에 잘 전달하여 표시하며 클라이언트에서 보내는 제어 신호를 레고 블록으로 작성된 온실 모델에 잘 전달하여 작동시키는 것을 볼 수 있었다.

참 고 문 헌

[1] 박종진 외, “리눅스 기반 임베디드시스템”, 대영사, 2004.
[2] J. Bentham, "TCP/IP LEAN Web Servers for Embedded Systems", CMPBooks, 2002.
[3] 최병욱 외, “임베디드 리눅스”, 홍릉과학출판사, 2003.
[4] V. Coroama, J. Bohn, F. Matten, "Living in a Smart Environment-Implications for the Coming Ubiquitous Information Society," IEEE SMC 2004, pp. 5633-5638, Oct., 10-13, 2004.
[5] M.D. Ercegovac and T. Lang, *Digital Systems and Hardward/Firmware Algorithms*, Wiley, 1985.
[6] 하원규, 김동원, 최남희, “유비쿼터스 총서, 유비쿼터스 IT 혁명과 제3공간,” 전자신문사, 2002. 11.
[7] 한백전자 기술연구소, “HBE-EMPOS III-P270으로 배우는 임베디드 리눅스 프로그래밍,” 한백전자 출판부, 2005.
[8] J.J. Labrosse, “MicroC/OS-II 실시간 커널,“ 에이콘출판사, 2005.
[9] Gonzales, "Microwave Transistor amplifiers Analysis and Design," 2nd Edition, 1997.

※ 본 논문은 2006년도 청운대학교 교내연구비로 수행되었습니다.

저자 소개

박 종 진(정회원)

- 1985년 연세대학교 전기공학과 공학사
- 1992년 연세대학교 대학원 전기공학과 공학석사
- 1997년 연세대학교 대학원 전자공학과 공학박사
- 2009년 현재 청운대학교 인터넷학과 교수.

<주관심분야 : 지능시스템, 임베디드시스템, 인터넷 프로그래밍>

최 규 석(종신회원)

제8권 제6호 참조

- 1987년 1월~1997년 1월 (주)데이콤 정보통신연구소 연구원 및 (주)SK텔레콤 중앙연구원 책임연구원 근무, 1997년 ~현재 청운대학교 컴퓨터학과 교수

<주관심분야 : 이동통신, 인공지능, 인공생명, 지능형 교통체계(ITS), 이동 컴퓨팅>