

# 무선랜을 이용한 원격 제어 임베디드 시스템

준회원 김 경 근\*, 정회원 정 원 수\*, 오 영 환\*

## Remote Control Embedded System using Wireless LAN

Kyung-Kuen Kim\* Associate Member, Won-Soo Jung\*, Young-Hwan Oh\* Regular Members

### 요 약

RC(Remote Control)는 27MHz, 40MHz, 75MHz대의 3가지 주파수 대역을 사용하는 무선 제어 방식이다. 정보통신의 발달로 무선 제어로부터 요구하는 서비스가 점차 늘어나면서 데이터 전송량이 적은 AVR 기반의 RC 무선 제어 방식은 한계를 나타내기 시작했다. 1990년대 후반부터 컴퓨팅(Computing), 통신(Communication), 가전 기기(Consumer Electronics)의 3C 기술들의 융합으로 등장한 임베디드 시스템은 일반 PC와는 달리 특정한 작업만을 위해 설계된 시스템으로 발전되어 왔으며 Ethernet 표준을 채택하여 특정 작업의 영역을 넘어 네트워크와의 연동으로 다양한 서비스를 지원하게 된다. RC에서 사용하는 대역폭보다 큰 대역폭을 사용할 수 있는 무선랜(WLAN)과의 연동을 위해 임베디드 시스템을 사용한다면 네트워크가 가능하지만 임베디드 시스템은 일반 PC처럼 자유롭게 네트워크를 하기 어려우며 이를 위한 운영체제 및 미들웨어(Middleware)를 필요로 한다.

본 논문에서는 X-HYPER Intel PXA255 ARM CPU를 기반으로 한 임베디드 리눅스 시스템과 무선랜 환경에서의 원격 제어를 위한 시스템을 구현하였다. RC제어를 위한 AVR모듈과 임베디드 리눅스 시스템을 기반으로 PCMCIA를 통한 무선랜 환경 구축 및 일반 PC와 모바일 환경에서 제어를 위한 GUI환경을 구현하였다.

**Key Words** : Embedded, ARM, WLAN, AVR

### ABSTRACT

RC(Remote Control) is wireless control technique using radio frequency 27MHz, 40MHz, 75MHz. As the development of information communication, service requirement from wireless control is gradually increasing, AVR-based RC wireless control technique which is limited low transmission data began to indicate the limits. From late 1990s, Embedded system raised by 3C that consist of computing, communication, consumer is developing as designed system which is just for specific working different from general PC, and as using Ethernet, supports various service by connecting with network. If the user uses embedded system for connecting to wireless LAN(WLAN) that can use larger bandwidth than use in RC, networking is possible: however, embedded system is difficult to networking as easy as general PC, and OS and middleware are needed for networking.

In this paper, we designed remote control system with WLAN using embedded linux system based on X-Hyper255 Intel PXA255 ARM CPU. The System implemented WLAN environment from PCMCIA and GUI at Desktop PC and mobile environment for control based on embedded linux system.

### I. 서 론

컴퓨터 및 소프트웨어와 네트워크를 포함하는 정

보통신 분야의 기술이 급격히 발전함에 따라 유비쿼터스 컴퓨팅 환경이 등장하여 언제 어디서든 자유롭게 네트워크를 통한 단말장치의 연결 및 제어

\* 광운대학교 전자통신공학과 통신망연구실(1981kkk@hanmail.net)  
논문번호 : 08012-0226, 접수일자 : 2008년 2월 26일

가 가능하게 되었다. 그러나 RF(Radio Frequency)를 사용하여 원격 제어를 하는 RC 시스템은 지정된 통신 주파수 영역 내에서만 제어가 가능하기 때문에 유비쿼터스 네트워크 환경에 적합하지 않다. 또한 RC 무선 제어 방식은 AM/FM 변조 방식의 RF통신으로 40m~70m로 제한된 통신반경을 제공하고 대부분 단방향 제어용 정보만을 송출하고 있으며 대역폭 또한 수 바이트(Byte) 정도의 제어용 패킷(packet)만 전송이 가능하기 때문에 이미지, 동영상 같은 대용량 정보를 보내는데 많은 어려움이 있다.

또한 RF통신과 기존 네트워크의 연동을 위해서는 RS-232 인터페이스를 통하여 단말기로부터 PC로 정보를 전송한 후 다시 PC를 통해 네트워크에 접속해야 한다. RC 시스템의 이러한 문제를 해결하기 위해서는 무선랜(Wireless LAN)환경 구축이 필요하다. 무선랜이란 1999년 9월에 RF 기술을 이용하는 고속 무선 랜 사양의 표준 규격인 IEEE 802.11b가 확정된 기술이며 100m 반경에서 대역폭은 11Mbps이다. 무선랜을 사용하여 RC시스템을 구성하여 네트워크를 한다면 11Mbps의 대역폭을 사용하여 큰 용량의 데이터 전송이 가능하고 단방향 제어 정보가 아닌 양방향 데이터 전송이 가능하게 된다. 또한 네트워크와 접속된 AP(Access Point)가 있는 곳이라면 거리에 상관없이 제어가 가능하다. 그러나 직접적으로 RC 제어모듈에 무선랜을 지원하기에는 많은 하드웨어적 제약 사항이 존재하기 때문에 AVR 시스템으로 구성된 RC 모듈과 임베디드 시스템을 연동하면 무선랜을 통한 네트워크가 가능하게 된다.

본 연구는 X-Hyper 255B 임베디드 보드를 기반으로 임베디드 리눅스 환경에서 무선랜 기술을 활용하여 AP(Access Point)가 설치되어진 무선 환경에서 원격 제어 시스템 및 모바일 단말기의 프로토타입(Prototype)을 구현하여 대용량 데이터 전송 및 양방향 데이터 전송을 가능하게 했다. 임베디드 리눅스 시스템 구축 후 AVR로 구성된 RC 제어 모듈을 제어하기 위해 미들웨어 및 PCMCIA를 지원하여 AP에 연동한 후, Qt 기반의 애플리케이션을 통하여 호스트 PC에 GUI환경을 구축하였다. 본 논문의 구성은 다음과 같다. 2장에서는 임베디드 시스템 개발환경에 대해 알아본다. 3장에서는 제안한 시스템 모델에 대해 알아본다. 4장에서는 구축한 시스템의 성능 평가에 대해 알아본다. 마지막으로 5장에서 논문의 결론을 내린다.

## II. 관련 이론

### 2.1 임베디드 리눅스 시스템

임베디드(Embedded)라는 단어 자체는 컴퓨터 시스템에서는 “내장된”이란 뜻으로 해석되고 있다. 즉, 컴퓨터 하드웨어와 소프트웨어가 조합되어 특정한 목적을 수행하는 시스템이 바로 임베디드 시스템(Embedded System)이다. 임베디드 시스템은 기존에 사용된 Micom 방식에 비하여 여러 가지 이점이 있다. 우선 다양한 기능을 더하거나 향상 시킬 때 비싼 하드웨어를 교체하는 것이 아니라 소프트웨어만 교체함으로써 제품 생산의 경비를 줄일 수 있으며 유지보수를 쉽게 할 수 있다. 특히 시스템을 개발하고 생산을 할때 단순히 하드웨어 위주로 개발된 Micom과 비교하면 제품을 모방하기가 쉽기 때문에 지적 재산권 문제가 발생할 수 있지만 임베디드 시스템의 경우 소프트웨어가 큰 역할을 담당하기 때문에 구성된 부분중 핵심 부분을 감출 수 있어 보호가 용이하다. 그림 1은 임베디드 시스템 모델을 나타내고 있다.

임베디드 시스템의 구성요소 중 하나인 임베디드 프로세서는 임베디드 장치내에 내장되는 마이크로프로세서를 지칭한다. 주로 제어와 계산을 수행하는 임베디드 프로세서는 기존에 사용되어진 마이크로컨트롤러(MCU)보다 성능은 높지만 범용 프로세서인 Intel이나 AMD 같은 범용 프로세서 보다는 낮은 성능을 가지고 있다. 하지만 그 적용 범위는 일반 PC 위주인 범용 프로세서 보다 넓다. 우선, 비트수를 살펴보면 8비트나 16비트가 주요 제품이었지만 최근들어 32비트 이상의 프로세서를 채택하는

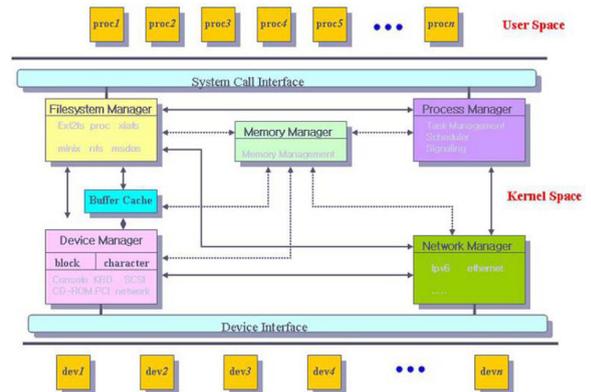


그림 1. 임베디드 시스템의 구성  
Fig. 1. Embedded System Reference Model

고성능 시스템이 늘어나고 있는 추세이며 프로세서에 메모리와 각종 보조회로들을 한 개의 칩에 탑재시켜 모든 기술을 제공할 수 있는 시스템 온 칩(SoC : System on Chip) 기술의 발달로 임베디드 시스템의 확산은 점점 가속되고 있다.

임베디드 시스템은 운영체제의 도움 없이 순차적인 Micromicro 프로그램만을 이용하여 사용자의 요구를 만족시키는 단순하며 순차적인 방식이었지만 최근의 임베디드 시스템은 시스템 자체에 대한 성능과 기능에 대한 요구가 커지고, 유비쿼터스 컴퓨팅의 등장으로 인하여 네트워크나 멀티미디어가 시스템에 필수적인 개념이 되면서 임베디드 시스템의 기능이 이전에 비해 매우 복잡해지고 다양화되었다. 따라서 시스템에서도 PC에서처럼 운영체제의 개념이 필요하게 되었으며 임베디드 시스템의 특성상 실시간이라는 요소를 만족해야 했으므로 실시간운영체제가 임베디드 시스템에 도입되었다.

실시간 시스템은 시간 제약성의 강약에 따라서 구분되고 이를 위한 실시간운영체제는 코드 실행 시간에 따라 정확히 관리하고, 시스템 자원을 관리 및 응용 프로그램 개발을 위한 일관된 기반을 제공하는 일종의 프로그램이다. 따라서 효율적인 실시간 운영체제는 이러한 요구 사항에 부합할 수 있는 특성을 가져야 한다. 다음의 표 1은 임베디드 운영체제의 기능별 비교를 나타내고 있다.

임베디드 시스템에 들어가는 운영체제는 초기에는 상용 실시간운영체제가 주로 이용되어 개발되었으나 리눅스의 성능과 안정성이 입증되고 다양한 프로세서들을 지원하기 시작하면서 리눅스를 활용한 응용 사례가 늘어나고 있다. 현재 리눅스가 지원하는 프

로세서는 인텔 계열, 알파, 파워 PC, MIPS ARM계열, SH계열을 지원하고 있다. 또한 각종 주변 장치들의 드라이버 개발도 용이하다. 본 논문의 시스템 역시 임베디드 리눅스를 기반으로 개발하였다.

임베디드 리눅스 시스템의 개발 시에는 일반적으로 커널, 디바이스 드라이버, 그리고 기본적인 GNU 컴파일러 툴 체인을 갖추어야 한다. 임베디드 소프트웨어의 개발은 운영체제 부분(OS porting), 장치 드라이버 개발, 미들웨어 개발, 그리고 GUI(Graphic User Interface)로 구성되어지는 응용프로그램 등 4개의 부분으로 구분되어진다. 이외에 일부 임베디드 리눅스 소프트웨어 개발 벤더들은 포괄적인 개발 툴들과 구성 요소들을 제공한다. 이 툴들에는 일반적으로 크로스 컴파일러, 디버거, 에디터, 구성 툴, 디바이스 드라이버, 파일 시스템, 네트워크 스택 및 유틸리티가 포함된다. 다음 그림 2는 임베디드 리눅스 시스템의 개발 환경을 보여주고 있다.

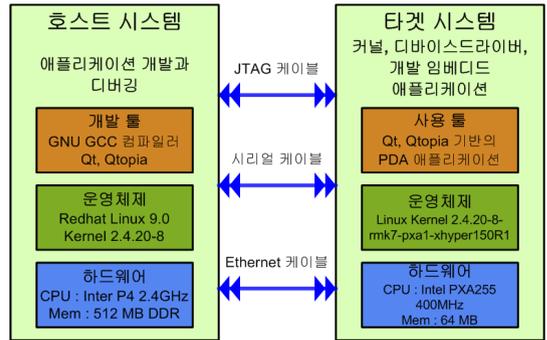


그림 2. 임베디드 시스템 개발환경  
Fig 2. Embedded System development dirgram

표 1. 임베디드 운영체제 비교  
Table 1. Comparing Embedded OS

	RT OS	PalmOS	Embedded Linux	Windows CE
실시간성	중간	불가	약함	약함
독립성	중간	불가	약함	약함
QoS	불가	불가	불가	약함
멀티미디어	불가	불가	약함	중간
범용성	약함	약함	중간	중간
크기	중간	약함	무거움	무거움
연결성	불가	불가	약함	중간
응용 분야	정보 가전	PDA	PDA 정보가전	PDA

## 2.2 임베디드 시스템에서의 무선랜

무선 랜(WLAN : Wireless Local Area Network)는 802.11의 표준에 바탕을 둔 기술로 2.4GHz 대역에서 운용되는 DSSS(Direct Sequence Spread Spectrum) 방식의 IEEE 802.11b 표준을 시작으로 다양한 표준이 개발되어 상용화 되고 있다. 다음 표 2는 상용화 되고 있는 무선랜 표준이다.

이 기술의 가장 큰 관건은 통신 속도에 관한 것으로 업계와 IEEE에서 고속의 무선 랜 기술에 대해 연구 개발을 하여 그 성과물로 가장 처음 나온 기술은 802.11b 이다. IEEE 802.11b는 미국전기전자학회(IEEE)가 1997년에 정한 무선랜 규격인 IEEE 802.11의 차세대 규격으로 IEEE 802.11b high rate 라고도 하며 2.4GHz대의 전파를 사용한다.

데이터 전송속도는 IEEE 802.11에서는 최고 2Mbps 였으나 802.11b에서는 11Mbps까지 올렸으며 IEEE 802.11과도 호환성이 있다. 802.11b의 특징은 다음과 같다.

- ◆ 2.4 GHz 대역의 주파수 확산방식을 사용한 무선랜 기술
- ◆ 2.4GHz대의 ISM 밴드에서 11Mbps까지의 속도를 제공
- ◆ 무선랜을 위한 표준으로서 물리계층 및 MAC (Medium Access Control) 계층을 정의

임베디드 시스템에서 802.11b를 사용하는 방식은 노트북과 PDA에서 무선랜 기능을 사용하기 위하여 주로 add-in 형태로 사용된 PCMCIA 카드를 임베디드 시스템에 추가로 장착하여 사용한다. 하지만 임베디드 보드마다 802.11b를 사용하기 위한 디바이스 드라이버의 표준이 명확치 않아 임베디드 제조회사마다 고유의 디바이스 드라이버를 사용하고 있다. 앞으로는 임베디드 기반의 Mobile PC에서 무선랜이 선택 사양이 아닌 필수 기능으로 선택될 경우, 무선랜 기능이 기기가 출시된 이후에 장착되는 애프터마켓(aftermarket) 형태 보다는 출시되는 시점에서 기기 내에 내장되는 Mini-PCI 또는 임베디드 솔루션이 주된 대체를 이룰 것으로 예상된다. 과거 무선랜 반도체 업체들의 가장 중요한 고객은 Cisco, Proxim, 3Com과 같은 무선랜 장비 업체들이었다. 그러나 앞으로 클라이언트 장치에서 무선랜 장치가 임베디드 솔루션 형태로 전환된다면 무선랜 반도체 업체들의 주된 고객은 무선랜 장비 업체들이 아닌 Dell, HP, IBM, NEC 등 주요 PC 업체들로 바뀌게 될 것이다. 따라서 이렇게 될 경우 동 산업에서

경쟁의 양상은 최신 기술을 선보이는 것도 물론 중요하겠지만 구매력이 높은 PC 업체를 고객으로 얼마나 많이 확보할 수 있는지 여부에 따라 크게 좌우될 것이다. 최근 Intel이 Centrino 플랫폼으로 주요 노트북 업체들을 공략하자, Intersil, Agere, Atheros 등 기존 무선랜 반도체 업체들은 802.11g, 802.11a와 같은 고속 솔루션이나 보다 진화된 듀얼 밴드 솔루션으로 PC 업체들을 고객으로 확보하기 위해 적극적으로 나서고 있다. Intel의 경우 현재 802.11b 제품만 출시하고 있으며 RF 트랜시버도 필립스의 제품을 이용하는 등 기술적으로도 성숙되지 않아, 기존 무선랜 반도체 업체들이 Intel과 본격적으로 경쟁하게 되기까지는 아직 다소간의 시간적 여유가 남아 있다고 볼 수 있다.

### 2.3 AVR(Alf Vergad RISC)

AVR은 미국의 ATMEl사에서 제공하는 CPU로서 RISC(Reduced Instruction Set Computer) 타입의 고속 8비트 마이크로컨트롤러이다. 마이크로컨트롤러란 하드웨어 제어용으로 최적화된 완전한 컴퓨터 시스템을 일컫는 것으로서 프로세서 전체, 메모리, 모든 입출력 주변장치를 한 조각의 실리콘 위에 포함하고 있다. 이미 RISC 구조를 채택했음에도 불구하고 같은 실리콘 조각 위에 있다는 것은 속도가 또 한번 향상됨을 의미하는데 이는 IC 밖에 있는 주변장치에 비해 안에 있는 주변장치에 쓰고 읽는데 시간이 적게 소요되기 때문이다.

AVR 프로세서의 메모리 부는 Harvard 모델을 기반으로 하고 있다. Harvard 모델에서는 메모리가 여러 부분으로 분리되어 있어 접근 시간이 빠르고 용량이 크다. AVR 프로세서는 FLASH 코드 메모리 영역, 데이터 메모리 영역, EEPROM 메모리 영역으로 분리되어 있다. 읽기/쓰기를 위한 데이터 메모리는 세 부분으로 분리되어 있으며, 가장 하위영역은 32개의 범용 작업용 레지스터이고, 그 다음은 64개의 입출력 레지스터이고, 그 다음은 내부 SRAM이다. 범용 작업용 레지스터는 프로그램 수행 중 사용되는 지역 변수와 기타 임시 데이터를 저장하기 위해 사용된다. 64개의 입출력 레지스터는 마이크로컨트롤러에 내장되어 있는 입출력 장치와 주변장치와의 인터페이스용이다. 그리고 내부 SRAM은 일반 변수를 저장할 영역으로 사용되며 프로세서의 스택으로도 사용한다. 즉, 내부 SRAM의 일부를 스택으로 사용한다. 그림 3은 데이터 메모리 영역의 메모리 맵이다.

표 2. IEEE 802.11 표준안  
Table 2. Standard of IEEE 802.11 series

	802.11	802.11a	802.11b	802.11g
어플리케이션	Wireless data networking	Broadband LAN Access	Wireless Data Networking	Broadband LAN access
Spectrum	2.4GHz	5.15~5.25 5.25~5.35 5.725~5.825	2.4GHz	2.4GHz
Modulation Scheme	FHSS or DSSS	OFDM	DSSS	OFDM or DSSS
Data rate	1Mbps~2Mbps	54Mbps	11Mbps	54Mbps

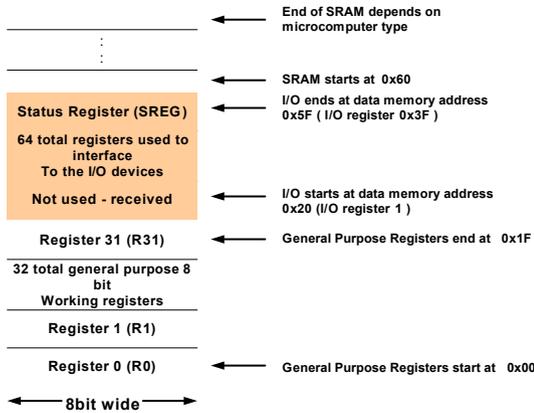


그림 3. AVR 데이터 메모리 구조  
Fig 3. AVR data memory map

ISP 케이블을 통해 PC의 Parallel 포트에 연결된 AVR 시스템은 Atmel C 언어로 작성된 코드를 AVR program Tool을 통하여 연결된 AVR 프로세서의 데이터 메모리로 전송되며 이를 바탕으로 하드웨어를 제어하게 된다.

AVR은 PIC(Peripheral Interface Controller)보다 높은 처리 속도를 가지고 있으며 8051보다 I/O 및 인터럽트(Interrupt)등의 데이터 처리에 필요한 자원이 많으며 중소규모의 산업용 제어기에 적합하다. AVR의 특징은 다음과 같다.

- ◆ ISP(In System Programming) 기능
- ◆ 단일 사이클 명령을 실행할 수 있는 RISC 구조
- ◆ 비휘발성 메모리 사용
- ◆ C언어 지원 및 처리
- ◆ 다양한 주변장치와 연결 가능
- ◆ 저전력 마이크로컨트롤러

### Ⅲ. 시스템 구성 환경

본 논문은 임베디드 리눅스 환경에서 무선랜 기술을 활용하여 원격 제어 시스템을 구현하였다. 그림 4는 제한한 시스템의 구성환경을 나타낸다.

무선랜에 접속하기 위해 임베디드 시스템과 결합한 RC 모듈은 AVR로 구성된 제어회로를 바탕으로 원격지에서 AP를 통하여 제어되며 RC 모듈에서 전송되는 영상을 미들웨어를 통하여 제공하게 된다. 또한 미들웨어를 통해 RC 모듈의 제어가 가능하다. RC 제어모듈에 사용한 임베디드 시스템은 X-Hyper255B를 기반으로 구성한다. 그림 5는 X-Hyper255B의 시스템의 구조를 나타낸다.

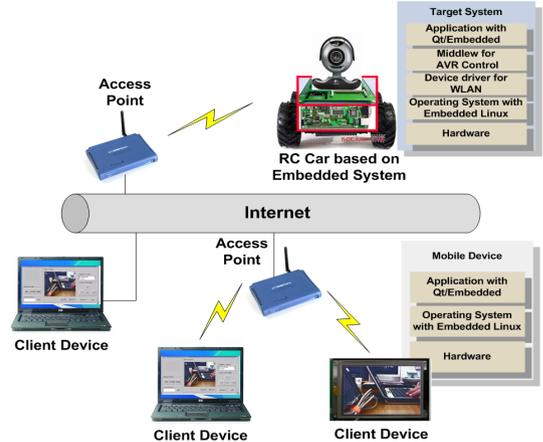


그림 4. 임베디드 시스템과 무선랜으로 구성된 RC 제어 환경  
Fig 4. RC car based on both embedded system and WLAN

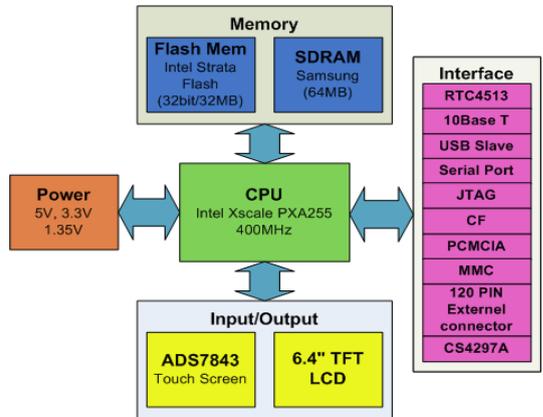


그림 5. X-Hyper255b 시스템 구조  
Fig 5. X-Hyper255b system block diagram

임베디드 리눅스 환경은 X-Hyper255B에서 직접적으로 이루어지는 것이 아니라 호스트 컴퓨터에서 이루어진다. 먼저 크로스컴파일 환경을 구축하여 호스트 컴퓨터에서 개발한 프로그램을 X-Hyper255B에 사용할 수 있도록 한다. 또한 호스트 컴퓨터인 리눅스 환경에서 Minicom 또는 tftp를 사용하여 운영 체제의 커널, 디바이스 드라이버, 어플리케이션을 개발한 후 이를 X-Hyper255B에 다운로드 한 후 동작시킨다. 다음은 X-Hyper255B기반 임베디드 리눅스 시스템 구축 과정이다.

- Toolchain 환경구축
- JTAG 환경구축
- Bootloader 환경구축
- Minicom 환경구축
- Bootp, tftp 환경구축

- 임베디드 리눅스 시스템 포팅 (Bootloader, Kernel compile, FileSystem)
- Qt 개발환경 구축, NFS 구축

구축된 임베디드 시스템과 RC 제어모듈의 연동을 위해 AVR 시스템을 사용하여 임베디드 시스템으로부터 받은 제어신호를 처리할 수 있는 미들웨어

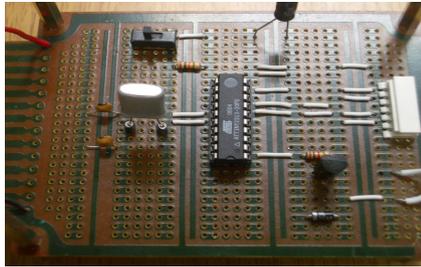
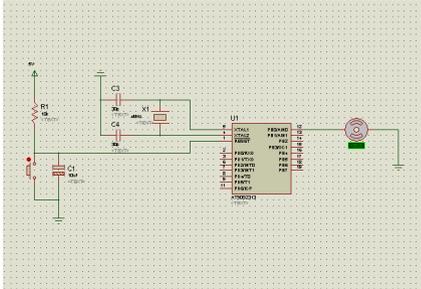


그림 6. AVR 제어모듈  
Fig 6. AVR control module

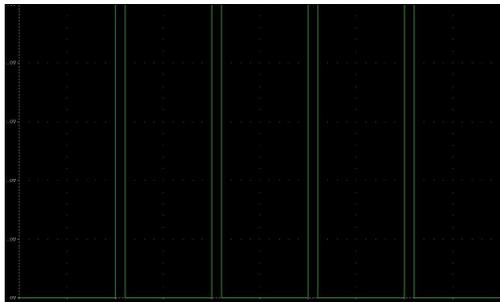
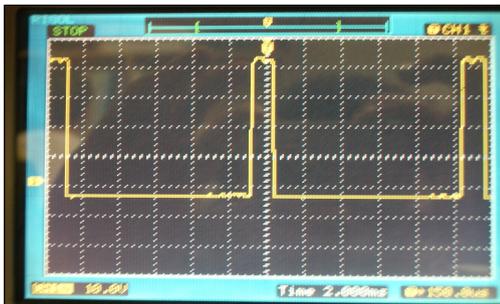


그림 7. Pspice와 오실로스코프의 출력신호 비교  
Fig 7. Comparing with Pspice and Oscilloscope

어가 필요하다. 그림 6은 X-Hyper255B에 구현한 임베디드 시스템과 RC 제어모듈과의 연동을 위해 제작한 AVR기반의 미들웨어이다.

AVR 미들웨어에서 입력받은 신호에 따라 올바른 신호의 출력이 이루어지는지를 확인하기 위해 오실로스코프와 pspice를 사용하여 신호의 검출을 확인하였다. 그림 7은 오실로스코프와 pspice에서 출력된 파형을 나타낸다.

또한 네트워크에 연결된 일반 PC와는 다른 모바일 환경에서의 제어를 실험하기 위해 X-Hyper255B를 기반으로 PDA 환경구현을 하였다. 그림 8은 X-Hyper255B를 기반으로 구성된 모바일 시스템이다.

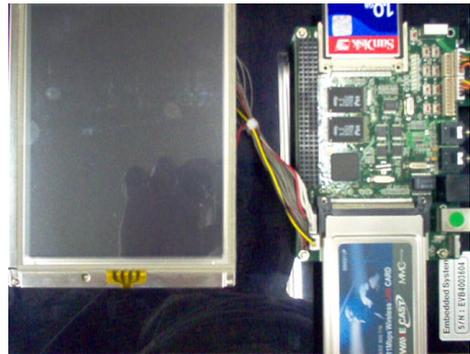


그림 8. 모바일 환경 구축을 위한 X-Hyper255B  
Fig 8. X-Hyper255B for Mobile environment

#### IV. 시스템 구현

본 논문에서 구현한 원격 제어 시스템은 제어 모듈로 구성된 AVR과 무선랜 접속 및 데이터 전송을 위한 임베디드 시스템 그리고 일반 PC 및 X-Hyper255B 시스템으로 구현한 모바일 단말장치에서 사용할 제어용 GUI로 구성된다. 그림 9는 구현한 RC 시스템을 나타내고 있다.

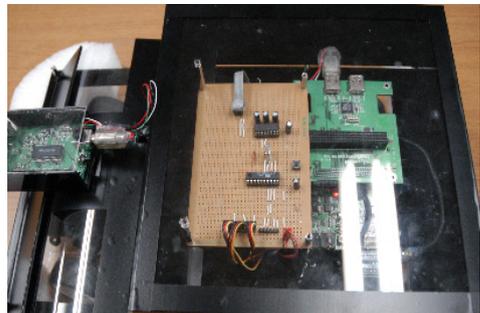


그림 9. 임베디드와 연동된 RC카  
Fig 9. RC car using embedded system

그림 10은 무선랜에서 RC 시스템의 성능 평가를 위한 환경을 나타낸다.

네트워크에 연결된 RC 시스템 제어 실험을 위해 다양한 OS에서 미들웨어를 제작하여 성능 테스트를 수행 했다. 그림 11은 리눅스 시스템에서 Qt 기반으로 구현한 GUI 화면이다.

그림 12는 Windows XP 환경에서 Visual C++ 6 MFC기반으로 구현한 GUI환경을 나타낸다.

그림 13은 X-Hyper255B 기반으로 구성된 Windows CE 환경에서 Embedded Visual C++ 4.0을 기반으로 구현한 GUI환경을 나타낸다.



그림 13. Windows CE 기반의 PDA 환경  
Fig 13. PDA prototype based on Windows CE



그림 10. 성능 평가를 위한 실험 시스템  
Fig 10. Experimentation system for performance evaluation

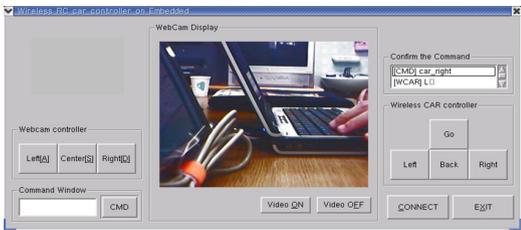


그림 11. 리눅스 환경에서의 제어용 GUI  
Fig 11. GUI for Linux



그림 12. 윈도우에서의 제어화면  
Fig 12. GUI for Windows XP

## V. 결론 및 향후과제

본 논문에서는 X-Hyper Intel PXA255 ARM CPU기반의 임베디드 리눅스 환경에서 무선랜 기술을 이용한 원격 제어 시스템을 구현하였다. 무선랜을 사용한 원격 제어 시스템을 네트워크에 접속되어 있는 다양한 시스템에서 미들웨어를 통하여 제어함으로써 실제 환경에서의 적용 가능성을 확인하였다.

AVR 제어 모듈을 구성하여 무선으로 RC카를 제어할 수 있는 환경을 구축하였으며, 무선랜 환경 구축 및 대용량 데이터 전송을 위한 임베디드 환경을 구축하여 단방향 데이터 전송 및 대용량 데이터를 전송하지 못하는 기존 RF 제어의 문제점을 해결하였다. 또한 다양한 운영체제 환경에서 제어용 GUI를 개발함으로써 네트워크에 접속된 다양한 환경에서의 제어 가능성을 확인하였다.

## 참 고 문 헌

- [1] IEEE Standard 802.16-2001 "IEEE Standard for Local and Metropolitan Area Networks-Part 16: A Air Interface for Broadband Wireless Access System," Apr.8 2002.
- [2] 박영환, "임베디드 시스템 & 임베디드 리눅스", 사이텍미디어 2002.
- [3] 박재호, "한빛미디어", 임베디드 리눅스 2004.
- [4] Korea Embedded Linux Project, <http://kelp.org>
- [5] Boa Webserver, <http://www.boa.org>
- [6] QT/Embedded, <http://www.trolltech.com>
- [7] GNU GCC, <http://gcc.gnu.org>

- 
- |   |   |     |
|---|---|-----|
| [8] 윤찬영외 “Wireless LAN환경에서 임베디드 리눅스기반으로 구현되어진 Mobile Remote Monitoring PDA Prototype 설계”, 대한의료 정보학회지 2004.                      | <b>김 경 근</b> (Kyung-Kuen Kim)<br>한국통신학회논문지 제32권 제12호 참조 | 준회원 |
| [9] 서성윤외 “임베디드 리눅스 기반의 홈네트워크 시스템 설계”, CICS 2004 .   | <b>정 원 수</b> (Won-Soo Jung)<br>한국통신학회논문지 제32권 제12호 참조   | 정회원 |
| [10] 장문석, 신광식, 정진하, 이양희, 심재홍, 이응혁, 최상방, 서비스 통합 시스템에서 지그비를 이용한 유비쿼터스 헬스케어 시스템의 설계 및 구현, 전자공학회논문지-TC 電會 第 TC編號, 16-24, 2006년 11월. | <b>오 영 환</b> (Young-Hwan Oh)<br>한국통신학회논문지 제32권 제12호 참조  | 정회원 |