

FPGA기반의 무선 온도 제어 시스템

박정욱[†], 고주영^{**}, 박종훈^{***}, 홍문호^{****}, 이영학^{*****}, 심재창^{*****}

요 약

본 연구는 ASIC기반의 유선 온습도 제어 시스템을 FPGA기반의 무선 제어 시스템으로 설계하고 구현하였다. FPGA 분야와 무선제어 시스템은 산업 현장에서 특히 온도와 습도 측정하는 분야에서 빠르게 성장하고 있다. FPGA는 저렴한 비용으로 빠르게 개발 할 수 있고, 원하는 CPU와 IO를 설정하여 시스템을 구성할 수 있다. 주변 장치를 IP 기반으로 설계하고 쉽게 변경할 수 있다. 무선화는 복잡하고 변화가 많은 현장에서 설치가 간편하고 유지 보수가 용이한 장점이 있다. 본 연구에서는 Sprtan-6 FPGA로 MicroBlaze 기반의 32비트 RISC CPU, 터치 및 주변 장치를 구현하고 임베디드 리눅스를 포팅하고, 지그비 무선 통신을 추가하였다. 시스템에 리눅스 OS와 웹 서버를 추가하여 웹을 통한 모니터링과 제어 기능을 구현하였다. 기존의 시스템과 비교할 때 성능향상 뿐만 아니라 개발이 쉽고 가격도 저렴하다. 연구에서는 특히 리눅스 기반의 개발 환경 구축과 사용자 인터페이스 구현에 연구를 집중하였다.

A Wireless Temperature Control System based on FPGA

Jeong Wook Park[†], Jooyoung Ko^{**}, Jong Hun Park^{***},
Mun Ho Hong^{****}, Yeung Hak Lee^{*****}, Jaechang Shim^{*****}

ABSTRACT

In this paper, we designed and built a wired temperature controller system which is based on ASIC for a wireless temperature controller system based on FPGA. FPGA devices and wireless controller systems are growing quickly especially for industrial systems for sensing temperature and humidity. FPGA can set up a desired system and a CPU, and directly set up or change a peripheral device based on an IP quickly for an affordable price. This wireless system is easy to install in the field where there are lots of changes and the system is complex. It also has advantages for maintenance. In this study, we are using a 32 bit RISC CPU based on MicroBlaze, with a touch interface, peripheral device, and porting the embedded Linux. Also, we added wireless communication using ZigBee. With this system we provide remote monitoring and control through the web by adding a web server. Compared to the original system, we say not only a performance improvement, but also more efficient development and cheaper costs. In this study, we focused especially on building a better development environment and a more effective user interface.

Key words: FPGA, temperature(온도), zigbee(지그비), microblaze(마이크로블레이즈), embedded system(임베디드 시스템), vmware(브이엠웨어)

※ 교신저자(Corresponding Author) : 심재창, 주소 : 경북 안동시 경동로 1375(송천동) 국립안동대학교(760-749), 전화: 054)820-5645, FAX : 054)820-6164, E-mail : jcshim@andong.ac.kr

접수일 : 2012년 1월 3일, 수정일: 2012년 3월 10일

완료일 : 2012년 5월 17일

[†] 준회원, 안동대학교 컴퓨터공학과 석사과정 (E-mail: ukihouse@naver.com)

^{**} 종신회원, 안동대학교 대학교육개발원, (E-mail: sonice@andong.ac.kr)

^{***} 정회원, 에이투스 대표이사

^{****} 정회원, 에이투스 연구원

^{*****} 정회원, 경운대학교 항공전자공학과 (E-mail: annaturu@ynu.ac.kr)

^{*****} 종신회원, 안동대학교 컴퓨터공학과

※ 본 연구는 중소기업청에서 지원하는 2010년도 산학연공동기술개발사업(No. 00042261)의 연구수행으로 인한 결과물임을 밝힙니다.

1. 서 론

정밀 온도 및 습도 제어 시스템이 산업의 여러 방면에서 널리 활용되고 있다. 본 연구는 산업용 무선 온도 제어기의 무선기능의 추가와 개발 및 유지보수의 편리성을 위해서 FPGA(Field Programmable Gate Array) 기반 시스템으로 설계 및 구현을 하였다. (주)삼원테크의 온도 제어기는 ARM 계열의 MPU(Multi-Processor Unit)가 활용되며 온도 측정 장치와 제어기는 RS 232C 및 485 방식의 유선기반의 방식이다. FPGA를 활용하여 제품의 성능은 향상하고 비용은 낮추며, 무선화를 통해서 설치 및 관리를 편리하도록 시스템을 설계하고 구현하였다.

현재 무선 제어 및 센싱 시스템은 산업현장에서 빠르게 성장하고 있으며, 네트워크기반의 통신 분야에서 적용이 확대되고 있다. 특히 산업현장에서 적용 사례가 많아지고 있는데 복잡하고 변화가 많은 현장에서 기존의 시스템보다 무선 제어 및 센싱 시스템은 설치가 간편하고 유지 보수가 용이한 장점이 있다. 본 연구에서는 표 1과 같이 하드웨어의 구현은 최신 FPGA 프로세서인 Spatran 6 기반으로 제어기를 설계하고, MicroBlaze 기반의 32비트 RISC CPU를 구현하였다. FPGA 기반으로 설계 및 제작은 개발 기간을 단축하고 가격을 낮추며, 안정성을 높여 경쟁력을 높이는데 있다[1].

OS로는 공개 소스 리눅스를 설치하여 개발 및 관리의 편의성을 제공하였다. 그리고 지그비(Zigbee) 통신기술은 초소형, 저전력, 저가격 측면에서 우수한 특징을 가지고 있으므로, 무선 임베디드 FPGA 기반 산업 및 컨트롤러 시스템을 효율적으로 지원할 수 있는 무선 통신 기술이다. 지그비를 이용하여 무선 통신을 설계하고 구현하였으며 주된 센싱은 온도를 읽고 AD 변환하여 무선을 통해 디지털로 전송한다. 특히, 반경 30m내에서 250Kbps의 속도로 데이터를

전송하며 메시 네트워크 구조를 이용하면 하나의 무선 네트워크에 약 255대의 기기를 연결할 수 있는 확장성이 매우 우수한 기술이다[2].

무선 온도 제어 시스템의 개발 환경은 임베디드 소프트웨어 개발 환경인 컴파일 서버, 사용자 단말, 타깃시스템을 사용한다. 본 연구에서는 VMware을 이용하여 하나의 PC에서 컴파일 서버 OS와 사용자 단말 OS를 설치했다[3]. VMware의 장점은 이와 같이 여러 개의 OS를 설치 할 수 있고, 멀티부팅이 가능하여 한 컴퓨터에서 다시 부팅하지 않고 여러 개의 OS를 운영할 수 있다. 또한 VMware 툴로 두 OS간의 파일 이동 및 하드웨어나 인터넷 등 주변기기의 확장이 용이하여 개발하기에 편의성을 제공한다 [4-5].

본 연구에서는 FPGA기반 무선 온도 제어 시스템을 설계 및 구현한다. 디질런트사의 Atlys Spatran-6 FPGA 개발 보드로 프로토타입의 구성 및 테스트를 한다. 그리고 최신 FPGA 프로세서인 Spatran 6 기반으로 제어기를 설계하고, MicroBlaze 기반의 32비트 RISC CPU를 구현한다[6]. 그리고 자일링스사에서 제공하는 공개 소스 리눅스를 운영체제로 채택하여 포팅하고, 실제 무선 온도 제어 시스템을 구현하기 위한 GUI(Graphic User Interface)와 네트워크 및 웹 서버애플리케이션을 구현한다[8]. 무선 온도 제어 시스템을 구성한 후에 기존의 시스템에 비하여 가질 수 있는 장점과 성능 비교에 관하여 고찰한다.

2. 관련 연구

2.1 FPGA

FPGA를 활용하는 논문으로는 류재경 등[1]의 SURF 알고리즘 기반 특징점 추출기의 FPGA 설계에서는 프로젝트의 특성에 맞게 메모리 접근과 메모리 사용량을 줄이는 방법을 이용하는 등 하드웨어를

표 1. 기존 제품과 본 연구의 비교

현재의 제어기	본 연구의 제어기	현재의 통신 방식	본 연구의 통신
ARM 기반의 CPU 저속 메모리 저용량 Flash	FPGA(Spartan 6) +고속 DDR2 메모리 +중용량 Flash	유선	유선 +무선의 추가
주변장치 칩 터치패드 용 칩	FPGA에 포함	RS232C RS422, RS485	RS232C RS422, RS485 +지그비 무선통신

설계하였다. 이경학 등[7]의 재구성 가능한 통신 단말 플랫폼의 설계 및 구현에서는 다양한 통신 시스템과 각각의 무선 네트워크들 간의 쉽고 빠른 인터페이스를 보장하기 위해 재구성이 가능한 플랫폼의 구현을 위해서 FPGA를 사용하였다. FPGA는 직접 주변 기기를 설계할 수 있고, 최적화를 할 수 있어서 많이 이용되고 있다.

본 연구에서 활용한 FPGA는 현장에서 빠르게 구현하여 적용해 보는 설계 방식으로 저렴한 비용과 시간으로 원하는 칩을 만들 수 있고, CPU 및 주변장치들을 IP 기반으로 직접 설계할 수 있으며, 필요에 따라 변경이 가능하다.

2.2 무선 통신(지그비)

지그비와 관련된 연구로는 이석철 등[8]의 USN 기반의 지하역사 모니터링 시스템의 설계 및 개발에서 환경 측정을 위해 온/습도 센서에 지그비 기반의 센서 네트워크를 통해서 데이터 수집을 한다. 센서 장치의 이동성과 네트워크 구현이 쉬워 센서 네트워크에서 많이 사용되고 있다. WPAN(Wireless Personal Area Network) 기술을 정의 IEEE 802.15.4 표준을 기반으로 하는 지그비는 저속 근거리 무선통신 기술로 2.4GHz 대역과 868/915MHz 대역을 사용하는 2개의 주파수 할당대역에서 동작한다. 저전력의 지그비 송수신기를 빛이나 기온, 압력과 같은 다양한 기능의 센서들과 결합하여 보다 큰 규모의 센서네트워크를 구성할 수 있는 기술이다. 본 연구에서는 지그비 무선 통신을 위해 펌텍사의 지그비 모듈을 설계 및 제작에 사용하였다. 여러 개의 센싱 노드와 데이터를 모으는 타깃 보드에 지그비를 부착하고 통신을 한다[9-10].

2.3 그래픽 라이브러리(nano-x)

이기훈[11]의 임베디드 리눅스 환경에서 디지털 TV에 최적화된 GUI 라이브러리 플랫폼 개발에는 GUI라이브러리로 마이크로 윈도우를 사용한다. 오픈 소스이며 경량화 되어 임베디드 시스템의 GUI로 적합하다. 현재는 Nano-X 윈도우 시스템 이라고 불리는 마이크로 윈도우는 공개 소스 프로젝트의 일환으로 MS 윈도우나 X-윈도우 같은 그래픽 윈도우 시스템을 지향하고 있다. 하지만 위에 열거한 윈도우 시스템보다 적은 용량과 적은 메모리를 필요로 하기

때문에 임베디드 환경을 위한 플랫폼이라고 할 수 있다. 현재는 포켓PC 등의 플랫폼으로 사용되고 있다[12].

이 플랫폼의 장점으로는 GPL 라이선스를 따르는 무료 공개 소스 플랫폼이며 독자적인 그래픽 엔진을 통한 디바이스 독립적인 특성을 가지고 있어 제작 어플리케이션의 확장이 자유롭다. 또한 서버/클라이언트 방식을 지원하여 멀티태스킹 기능이 가능하고 컴파일 시 수백 KB의 작은 용량만을 필요로 하기 때문에 메모리의 활용도가 더욱 높아지게 된다. 마이크로 윈도우는 Nano-X API와 함께 WIN32 API 와도 유사한 API를 지원함으로써 X-Window 프로그래밍에 익숙한 개발자와 WIN32 API 프로그래밍에 익숙한 개발자 모두가 쉽게 개발 할 수 있는 환경을 제공한다. 또한 기본 바이너리(Native Binary) 형식의 파일을 지원하고 있어 자바 가상 기계(Java Virtual Machine)에서의 구현속도보다 빠른 실행을 할 수 있다. 현재 마이크로윈도우에서 제공하는 장치(Widget)는 없지만 FLTK의 마이크로윈도우용인 FLNK와 함께 Nano-X 기반의 TinyWidget이 지원되고 있다[13]. 본 연구에서는 Nano-X 기반의 라이브러리를 사용하였다.

3. 구현 및 고찰

3.1 임베디드 실험 키트의 구성

본 연구에서 보드를 개발하기 전에 테스트를 위해 사용한 보드는 그림 1과 같은 Digilent 사의 Atlys Spartan-6 FPGA 개발 보드이다. Atlys 회로 보드는 완제품으로 Xilinx spartan-6 LX45 FPGA를 기반으로 바로 사용 가능한 디지털 회로 개발 플랫폼이다

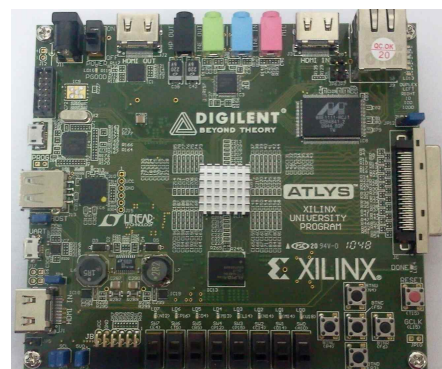


그림 1. atlys board

[20]. Gbit 이더넷, HDMI 비디오, 128Mbyte DDR2 메모리 어레이, 오디오 그리고 USB 포트를 포함한 고급의 주변기기 들을 포함하고 있고, Xilinx 마이크로브레이즈와 같은 디지털 시스템에 이상적으로 호스트가 된다[6,18]. 다음의 그림 1에는 JPEG을 구현하는 전체의 블록선도가 제시되어 있다[4].

테스트를 위한 개발 환경의 구성은 그림 2처럼 노트북, HDMI 용 모니터 그리고 Alys 보드로 구성된 다. 이 테스트에서는 센싱을 위한 무선 통신(지그비)과 TFT LCD 테스트를 제외하고, 임베디드 리눅스, 포팅(porting) 그리고 GUI(Graphic user interface)를 위한 테스트를 하였다[5-17].

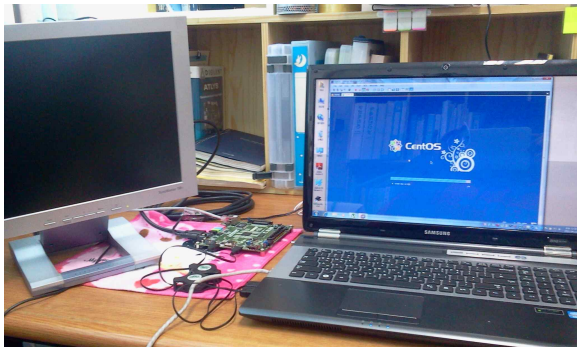


그림 2. 테스트 개발 환경

3.2 하드웨어

본 연구에서는 무선 온도 제어시스템을 위한 하드웨어를 직접 설계하였다. 자일링스사의 sptran-6 FPGA칩을 사용하고 기존의 삼원 온도 시스템을 구현할 주변 장치와 무선 통신에 필요한 지그비 모듈을 설계하였다. 그림 3과 그림 4는 설계한 보드의 실제



그림 3. 개발된 시스템 앞면 영상



그림 4. 개발된 시스템 뒷면영상

모습이다.

그림 5은 개발된 지그비 모듈 센서의 그림이다. 그림처럼 지그비 모듈에 온도 센서가 장착되어 있다. 이 센서에서 온도를 읽어 지그비 모듈은 타깃보드로 데이터를 송신하게 된다.

본 연구에서는 H/W 설계에 있어서 타깃시스템의 명세는 그림 6과 같다. ① CPU 로직은 32비트 RISC 구조이며, Xilinx MicroBlaze 7을 활용하였다[19]. MicroBlaze는 Spatran 6에 완전히 삽입되어 추가 지원기기가 요구되지 않도록 하였다. 데이터의 처리를 위해 고속의 외부 메모리로 DDR2를 추가하였다. 하드웨어 디버그와 프로그램 다운로드를 위해서 JTAG Debug 인터페이스를 사용하였다. ② 데이터 입출력으로는 16550기반의 Serial UART를 사용하였고, 네트워크로 이더넷 컨트롤러를 연결하였다. OS는 산업용 컨트롤러에 적합하도록 리눅스 기반으로 프로그래밍 환경을 구성하였다. ④사용자 인터페이스를 위해 그래픽은 Xilinx TFT 그래픽 컨트롤러를 사용하였다. ⑤ 데이터 및 분석 자료의 저장을 위해 SD Card를 장착하였다. 그리고 Firmtech의 Mango 기반 지그비 무선 통신의 설계를 하고 온도

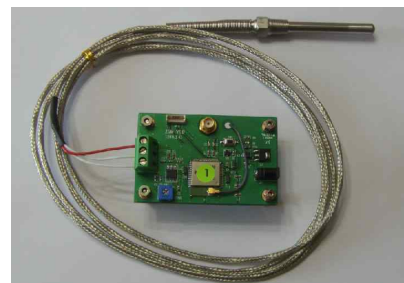


그림 5. 온도 측정 지그비 모듈

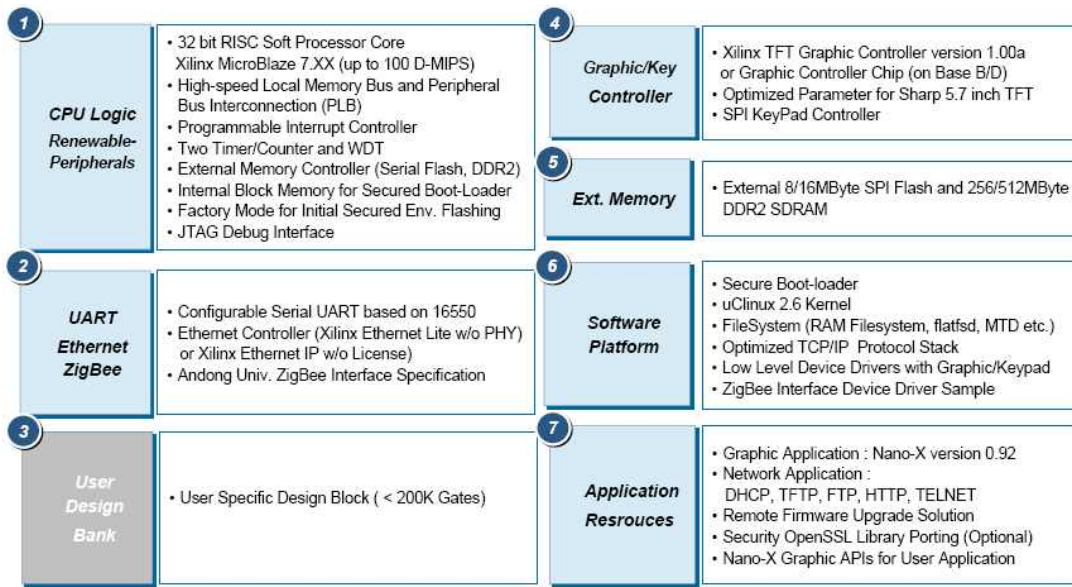


그림 6. 하드웨어 명세

센서를 부착하여 온도를 제어 할 수 있도록 하였다.

Xilinx 하드웨어마다 BSP(Board Support Package)를 설치하여 타깃 보드에 맞는 설정을 생성한다. BSP는 각 보드 제조사를 통해 제공 받을 수 있다. 현재 Xilinx OSL에서는 BSP 대신 디바이스 트리 방식을 사용하여 플랫폼에 상관없이 하드웨어 정보를 자동으로 생성한다. 이렇게 만들어진 디바이스 트리

는 부팅시 메모리, 플래시, 시리얼, 이더넷 등 하드웨어에 대한 정보를 자동으로 설정하는데 사용된다.

그림 7는 IP 블록 다이어그램으로 본 연구의 시스템의 IP들을 구성해 놓은 것이다. xilinx사에서 제공하는 xilinx ISE tool(ISE 12.4버전)을 사용해서 각 IP에 해당되는 설정을 완료한다[20]. 표 2는 설정된 IP들의 주소 맵이다.

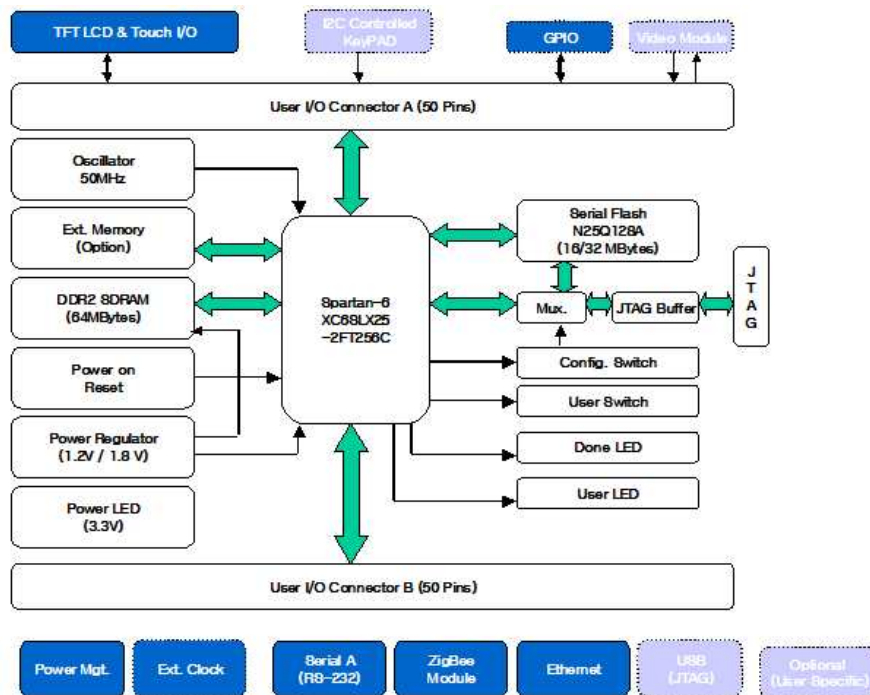


그림 7. IP Block Diagram

표 2. Address Map

Name	Base Address	High Address	Size(Byte)	Instance
BRAM	0x00000000	0x00003FFF	16K	lmb_cntlr
DDR2	0x48000000	0x4BFFFFFF	64M	MCB_DDR2
Interrupts	0x81800000	0x8180FFFF	64K	xps_intc_0
Console	0x83E00000	0x83E0FFFF	64K	xps_uart16550_0
Ethernet	0x83E80000	0x83E8FFFF	64K	xps_ethernetlite_0
MDM	0x84400000	0x8440FFFF	64K	mdm_0
ZIGBEE	0x84500000	0x8450FFFF	64K	xps_uart16550_1
TFT	0x85100000	0x8510FFFF	64K	xps_tft_0

3.3 소프트웨어

3.3.1 임베디드 운영체제의 선정

임베디드 운영체제는 타깃 시스템인 하드웨어에 내장 되어 동작하는 소프트웨어를 의미한다. 점차 임베디드 시스템의 요구기능이 복잡해짐에 따라 프로그램도 복잡해지고, 지원해야 하는 서비스가 다양화됨에 따라 프로그램을 보다 효율적으로 관리하기 위해서 최근의 모든 임베디드 시스템에서는 임베디드 운영체제를 사용한다. 현재 임베디드 운영체제의 시장은 실시간 운영체제로는 WindRiver 시스템의 VxWorks가 가장 많이 사용되고, 비실시간 운영체제로는 PalmOS, 윈도우 CE, 임베디드 리눅스 등이 있는데, 윈도우 CE는 PDA 및 일반가전 시장에서 무서운 속도로 성장하고 있는 추세이다. 그러나 윈도우 CE는 제품 사용에 대한 라이선스 비용이 높고, 메모리 관리나 프로그램 사이즈 그리고 속도 면에서 많은 단점을 가지고 있기 때문에 많은 개발자들이 대안을 찾고 있으며, 그 대안으로 임베디드 리눅스가 대두되고 있다. 임베디드 리눅스는 타깃 시스템이 되는 프로세서의 종류에 따라 크게 두 가지의 버전으로 발전되어 왔는데 일반적으로 리눅스는 가상메모리라는 기법을 지원하는 MMU(Memory Management Unit)가 있는 프로세서에서 사용될 수 있다. 하지만 MMU가 없는 프로세서에서는 일반적인 리눅스를 사용할 수 없기 때문에 이러한 MMU를 내포하지 않는 마이크로프로세서를 지원하기 위해 기존의 리눅스를 기반으로 해서 새로운 임베디드 리눅스인 uClinux이다. 일반 리눅스에 비해 안정성 및 네트워킹 기능이 떨어지는 단점을 가지나, 커널 사이즈가 기존의 리눅스에 비해 작고 경량화 된 장점을 지니기 때문에

MMU를 지원하지 않는 임베디드 시스템에는 적합하다고 할 수 있다. 개발될 시스템에서는 MMU를 지원하는 프로세서이므로 일반적인 리눅스 운영체제를 선택하였다[21].

Microblaze는 자일링스에서 제공하는 32bit RISC soft-processor이다. Microblaze의 장점은 자신의 용도에 맞게 custom IP의 연결이 쉽고 필요에 따라 최적화가 가능하다. 무료로 제공되어 별도의 라이선스가 필요 없다. Microblaze를 지원하는 상용 OS로는 linuxworks 사의 BlueCat과 Petalogix의 PetaLinux가 있다. 공개 소스 OS로는 자일링스에서 운영하는 공개 소스 리눅스가 있다. 본 연구에서는 자일링스의 공개 소스 리눅스 2.6 kernel으로 구현되었다.

3.3.2 교차 개발환경 및 포팅

타깃 시스템에 수행될 운영체제 및 프로그램의 컴파일을 위해서는 교차개발 환경을 구축해야 한다. 교차개발 환경이라 함은 주 컴퓨터(host computer)와 타깃 시스템의 프로세서가 서로 다른 경우, 주 컴퓨터에서 사용되는 타깃용 도구라고 정의할 수 있다 [15]. 교차개발 환경을 이용하여 컴파일 된 프로그램은 JTAG, 이더넷 또는 시리얼 포트를 통해 타깃 시스템의 플래시 메모리와 같은 저장장치로 다운로드 된다. 기본적으로 임베디드 시스템에 사용될 임베디드 리눅스를 컴파일 하기 위한 교차개발 환경을 구성하기 위해서는 교차 컴파일러(cross compiler), 교차 링커(cross linker), 바이너리 도구(binary utility), 그리고 교차 컴파일러가 컴파일시 참조하는 라이브러리(cross library)가 필요하다. 또한 이러한 교차 개발 환경이 구성되는 주 컴퓨터의 운영체제로는 리눅스를 주로 사용되는데, 사용자에 따라서 윈도우즈 상

에 이러한 교차 개발 환경을 사용하기 위해 Redhat사에서 제공되는 cygwin이라는 개발 환경을 사용하기도 한다. 호스트 시스템에서는 이 같은 tool을 통해 타깃 시스템에서 실행될 수 있는 실행 코드를 만들어 낸다. 본 연구에서는 교차 개발 컴파일러 환경(tool chain)을 위해서 자일링스에서 제공하는 공개 소스 리눅스를 사용한다. 컴파일러로는 자일링스에서 제공하는 microblaze 4.1.2 버전의 gcc 컴파일러를 사용한다. 공개 소스 리눅스 플랫폼을 사용하기 위해서는 표 3과 같은 툴과 라이브러리가 필요하다.

자일링스에서는 운영하는 지트 서버(git server)로부터 리눅스 커널, 크로스-컴파일러, 부트로더 그리고 비지박스 등을 다운로드 받을 수 있다. 지트 서버의 주소는 git://git.xilinx.com 이다. 각 소스 다운로드 방법은 다음과 같다.

```
Cross compiler: [PC~]$
git clone git://git.xilinx.com/xldk/microblaze_v1.0.git

Linux kernel: [PC~]$
git clone git://git.xilinx.com/linux-2.6-xlnx.git

Bootloader: [PC~]$
git clone git://git.xilinx.com/u-boot-xlnx.git

BusyBox: [PC~]$
git clone git://git.xilinx.com/busybox.git

Device Tree : [PC~]$
git clone git://git.xilinx.com/device-tree.git
```

표 3. 공개 소스 리눅스 플랫폼을 사용하기 위한 툴 및 라이브러리

Tool / Library	YUM/RPM package for RHEL/CentOS/Fedora	APT Package for Debian/Ubuntu
dos2unix	dos2unix	tofrodos
gcc	gcc	gcc
git	git	git-core
gdg	gnupg	gnupg
make	gnutls-devel	make
netstat	net-tools	net-tools
ncurses	ncurses-devel	ncurses-devel
tftp server	tftp-server	tftpd
zlib	zlib-devel	zlib-dev

RedHat/CentOS/Fedora 타입의 시스템의 경우 'yum' package manager를 사용한다.

```
$ yum install -y <yum-package-name>
```

Debian/Ubuntu 타입의 시스템의 경우 'apt' package manager를 사용한다.

```
$ apt-get install -y <apt-package-name>
```

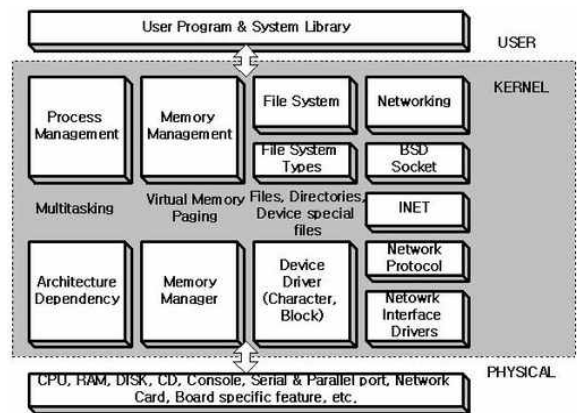


그림 8. 리눅스 커널의 구조

리눅스 커널은 그림 8와 같이 하드웨어 계층 위에 존재하며 마이크로프로세서에 밀접하게 관련된 부분과 관련이 없는 부분으로 나뉘어진다. 리눅스 커널은 많은 컴포넌트들로 구성되며 각각은 자신이 사용하는 데이터 구조와 연산들로 이루어져 있다. 각 블록들은 상호 독립적으로 구현되어 있으며 전체적으로 상호 유기적으로 결합된다. 이러한 커널 위에 시스템 라이브러리와 응용프로그램이 위치하는데 실질적으로 커널은 일종의 라이브러리와 같은 것으로 사용자 응용프로그램이 요청하는 서비스에 응답하는 것이지 새로운 무엇인가를 생성하여 수행하는 것이다. 그리고 이것은 사용자 응용 프로그램의 요청에 따라 하드웨어에 직접적으로 접근하여 특권 명령어(privileged instruction)를 사용할 수 있도록 서비스

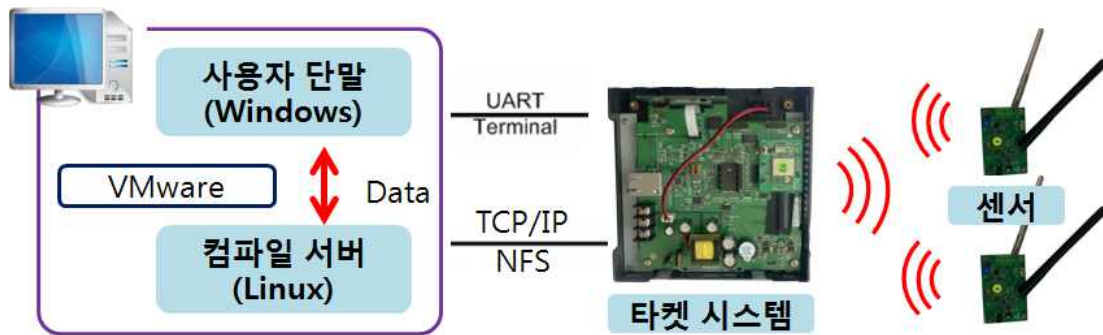


그림 9. 임베디드 시스템의 개요도

를 제공한다. 일반적으로 리눅스 포팅(porting)이라는 것은 하드웨어에 의존적인 부분을 수정하는 것을 의미하며, 이는 하드웨어 의존적인 코드인 마이크로프로세서 및 타겟 보드 초기화 코드 등이 포함된다.

그림 9는 본 연구에서 구성할 FPGA 기반의 온도 제어 시스템의 개발 환경을 보여준다. 이들은 전형적인 임베디드 시스템의 구성에서 컴파일 서버와 사용자 단말을 VMware 툴을 이용하여 하나의 PC안에 구성한다. 그래서 하나의 OS에는 익숙한 윈도우 환경을 사용하면서 자일링스사의 ISE Design Suite을 이용하여 MicroBlaze 기반의 CPU, DDR2, Flash 및 각종 입출력 하드웨어 장치를 설계하고 JTAG를 이용하여 타겟시스템으로 다운로드한다. 다른 OS에는 컴파일 서버로 타겟시스템에서 이용할 텔넷, NFS (Network File system) 등 네트워크 서비스 및 소스 코드 편집, 교차 개발 환경을 구성해서 컴파일을 진행한다. VMware를 사용하여 따로 컴파일 서버와 사용자 단말의 데이터 통신을 위한 네트워크(samba)를 구성할 필요가 없고 VMware tool을 이용하여 드래그-드롭(drag-drop) 등 간단하게 파일을 이동할 수 있다. 또 개발환경의 버전관리와 추가기능에 대한 확장이 쉬워져서 개발하는데 용이하다. 본 논문에서는 VMware 환경에서 CentOS 5 기준으로 사용되었다[10].

그리고 개발을 진행하면서 일반적으로 컴파일 결과물을 매번 다운로드 후 타겟시스템에서 실행하는 것은 수분이 걸리므로 매우 비효율적인 방법이다. 텔넷은 컴파일 서버에서 타겟시스템의 프로그램을 실행하고 컴파일 할 때 사용한다. NFS는 컴파일서버와 타겟리눅스 시스템간의 파일 시스템 공유 서비스를 제공한다. 그래서 개발을 하는 평상시에는 타겟 시스템에서 NFS로 컴파일 서버에 있는 실행파일을 원

격으로 실행해서 테스트한다. 개발 초반에 다소 번거롭지만 개발의 용이함을 위해 구축해야하는 필수 요소이다[17].

그래서 본 연구에서는 개발을 좀 더 용이하게 하기 위해서 VMware를 이용해서 윈도우에서 이용 가능한 소프트웨어와 리눅스에서 활용 가능한 텔넷, NFS 서비스 등의 개발환경을 하나의 PC에서 사용하였다.

3.4 FPGA기반의 무선 온도 시스템 구현

본 연구에서 개발된 보드를 이용해서 무선 온도 제어 시스템을 구현했다. 무선 온도 제어 시스템은 터치스크린(TFT LCD)의 인터페이스를 이용한 제품 조작 및 설정이 가능하고 정밀 측정 가능한 온도 센서 최대 255개로부터 값을 읽어 들여서 그 값들을 모니터링, 분석, 제어가 가능하다. 그림 10은 무선 온도 제어 시스템의 센서로부터 온도 값을 받아 모니터링 해주는 화면을 구현한 그림이다. 센서는 주기적으로 값을 읽고 설정된 시간만큼 주기적으로 전달한다. 각 센서 노드에서 지그비로 전달한 값을 디스플레이 한다.



그림 10. 시스템 실행화면

그리고 시스템의 웹서버를 이용해서 사용자는 웹을 통해 모니터링 및 제어를 할 수 있다. 시스템은 받아온 값을 웹에서 출력하고 온도의 설정 및 시스템의 상태 설정 등의 제어를 제공한다.

3.5 고찰

본 연구에서는 ASIC 기반의 시스템에서 ASIC 보다 진보된 기술인 FPGA로 하나의 칩에 MCU+Flash+I/O를 포함하고, 여러 개의 프로세서를 장착할 수 있고 언제든지 바꿀 수 있으며, 각종 IO를 내장할 수 있다. 그리고 FPGA칩 설계기술은 제품단가를 현저히 낮출 수 있고, 개발 기간을 기존 방식 보다 약 1/5로 줄일 수 있으며, 시스템을 소형화할 수 있는 기술이다.

RS485 방식의 통신에서 무선 통신을 더함으로써 유선이 가지는 단점인 접속할 수 있는 양과 설치의 어려움을 해소할 수 있었다. 그리고 지그비 무선 통신기술은 초소형, 저전력, 저가격 측면에서 우수한 특징을 가지고 있으며, 유선 방식의 32개의 연결 피드를 최대 255개로 연결 가능하다. 그리고 웹 서버를 구현함으로써 기존의 단말에서 모니터링과 제어를 웹상으로도 가능하게 하였다.

본 논문에서 구현된 플랫폼인 임베디드 리눅스 OS는 동일한 OS에서 동작하는 다른 기종과도 이식 가능하고 융합이 용이하다. 그리고 플랫폼 내부에 존

재하는 웹 서버와 CGI통신으로 인터넷 원격 접속 및 제어가 가능하고 센서/제어 모듈은 플랫폼 내부에 동기화 하여 따로 동작 혹은 Stand alone 모드로 동작 가능하다.

시스템을 구현할 때 고려해야 할 사항은 그래픽 라이브러리인 nano-x는 적은 메모리를 사용한다는 장점이 있지만, 그만큼 개발자가 시스템을 구현하기 위해서 직접 개발해야할 부분이 많다. 특히나 폰트 및 한글화 작업과 이미지 및 그리기, 대화상자 부분에서는 따로 구현해 주어야 한다. 그래서 FLTK 등과 같은 라이브러리가 지원되고 있지만 따로 추가 포팅 작업이 이루어 져야 한다. 그리고 지그비 통신에서 50개 이상의 노드와 통신을 할 경우에 동 시간에 통신을 할 경우에 충돌이나 간섭이 일어나는 경우가 있는데, 이를 위해 네트워크 구성 및 알고리즘이 필요하다.

4. 결 론

본 연구에서는 산업용으로 활용될 무선 임베디드 시스템을 FPGA 기반으로 설계하고 구축하고 지그비 무선 통신을 추가하였다. Spartan 6 FPGA에 32비트 임베디드 프로세서인 MicroBlaze를 탑재하여 제어 시스템을 구현하여 관리의 편리성과 안정성을 높였다. 리눅스 OS 기반에서 제어 프로그래밍을 구현하였다. 센서와 제어기, 제어기와 컨트롤러 사이에는 근거리 무선통신인 지그비를 적용하여 무선화 함으로 설치와 관리가 용이하다. 또한 제어 시스템을 FPGA 기반으로 구현함으로써 시스템의 안정성이 높았다. 본 연구는 개발 환경을 VMware를 이용하여 개발을 기존의 개발 시스템보다 개발을 용이하게 하였다. 그리고 그래픽 지원을 위해 nano-X를 활용하였으며, 다른 그래픽 라이브러리보다 사이즈가 작아, 임베디드의 제한적 자원에 효율적이다. 앞으로의 향후 연구과제는 현재의 시스템의 구현을 모듈화 하고 최적화된 라이브러리를 만드는 것이고, 웹 서버의 확장 및 보안에 대해서 연구하고, 스마트 폰과의 연계를 통해서 기기를 통해서 모니터링 및 제어를 구현할 계획이다.

참 고 문 헌

[1] 류재경, 이수현, 정용진, "SURF 알고리즘 기반



그림 11. 모니터링 및 분석



그림 12. 웹을 이용한 모니터링

- 특징점 추출기의 FPGA 설계,” 멀티미디어학회 논문지, Vol. 14, No. 3, pp. 368-377, 2011.
- [2] 심재창, 김익동, 지그비기술의 응용과 실습, 홍릉과학출판사, 안동, 2007.
- [3] 나종화, 강순주, 윤용익, 반윤용, 은성배, 김홍남, Embedded System Programming 이론 및 실습, 한국전자통신연구원, 대전, 2003.
- [4] 임채덕, 김태호, “임베디드 소프트웨어 통합 개발 환경 기술,” 정보과학회지, Vol. 24, No. 8, pp. 46-52, 2006.
- [5] 남원길, 통합 브이엠웨어 서버 시스템 구축을 통한 비용절감 방법, 숭실대학교 정보과학대학원 석사학위논문, 2009.
- [6] 윤성기, 이규선, 강병권, “FPGA 및 마이크로프로세서를 적용한 무선컨트롤러 구현,” 춘계학술 발표대회논문집, Vol. 2004, No. 1, pp. 405-408, 2004.
- [7] 이경학, 고희화, “재구성 가능한 통신 단말 플랫폼의 설계 및 구현,” 멀티미디어학회논문지, Vol. 10, No. 1, pp. 66-73, 2007.
- [8] 이승열, 실시간 운영체제를 위한 내장형 웹서버 설계 및 구현, 충남대학교 대학원 석사학위논문, 2008.
- [9] 이석철, 정신일, 김창수, “USN 기반의 지하역사 모니터링 시스템의 설계 및 개발,” 멀티미디어학회논문지, Vol.12, No.11, pp. 1629-1639, 2009.
- [10] 심갑식, 장재혁, USN 기반의 무선 멀티 커뮤니케이션 회로 설계 및 구현, 한국콘텐츠학회 Vol. 11 No.8, 2011.
- [11] 이기훈, 임베디드 리눅스 환경에서 디지털 TV에 최적화된 GUI 라이브러리 플랫폼 개발, 경북대학교 일반대학원, 석사학위논문, 2007.
- [12] 김진태, 내장형 시스템을 위한 애플리케이션 플랫폼 구현, 성균관대학교 대학원, 석사학위논문, 2004.
- [13] 윤지훈, 채영훈, 문승진, “Nano-X Window System 기반의 모바일 플랫폼 개발을 위한 에뮬레이터 설계 및 구현에 관한 연구,” 한국지능시스템학회 학술발표 논문집, Vol.14, No.2, pp. 135-138, 2005.
- [14] 임철호, 임베디드 리눅스용 웹 브라우저 포팅, 한국기술교육대학교 대학원, 2004.
- [15] 강선례, 내장형 시스템용 웹 서버 개발, 상명대학교 대학원, 석사학위논문, 2004.
- [16] www.gnu.org/s/libmicrohttpd, GNU libmicrohttpd, 2011.
- [17] 조주현, “임베디드 실시간 시스템의 개발환경,” 한국정보처리학회지, Vol.9, No.1, pp. 120-126, 2002.
- [18] 최병윤, 임베디드 리눅스 웹 서버와 네트워크 디바이스 드라이버 설계 및 구현에 관한 연구, 안동대학교 대학원, 석사학위논문, 2004.
- [19] 정광성, 김창범, 문철홍 “xilinx spartan3 FPGA를 이용한 8bit processor LED 조명 board 설계,” 정보 및 제어 심포지엄 논문집, Vol. 9, No. 10, pp. 331-332, 2009.
- [20] <http://digilent.us>, Atlys Board, 2011.
- [21] 이재만, 이연옥, 이승호, SoC(system on chip)기반의 임베디드 실험 키트 및 응용 프로그램의 개발, 한밭대학교 논문집, Vol.23, No.- pp. 65-80, 2006.



박 정 옥

2011년 안동대학교 컴퓨터 공학과 졸업
2011년 3월~2012년 6월 안동대학교 컴퓨터 공학과 석사 재학중



고 주 영

1994년 호성여자대학교 의류학과 석사
2002년 국립안동대학교 멀티미디어공학전공 석사
2010년 국립안동대학교 멀티미디어공학전공 박사

관심분야 : 멀티미디어응용, 이러닝, 멀티미디어 콘텐츠

홍 문 호

ATUS 대표이사

박 종 훈

ATUS 연구원



심 재 창

1987년 2월, 1990년 2월, 1993년 8월 : 경북대학교 전자공학과 공학사, 석사, 박사
1994년~현재 국립안동대 컴퓨터공학과 교수
1997년~1999년 IBM T. J. Watson 연구소 연구원

2005년~2006년 미국 프린스턴대학교 Visiting Fellow Professor

관심분야 : 영상처리, 패턴인식, 컴퓨터비전, 임베디드 시스템, 무선 로봇

이 영 학



1988년 영남대학교 전자공학과 학사
1991년 영남대학교 전자공학과 석사
2003년 영남대학교 전자공학과 박사
2010년~현재 경운대학교 항공전자공학과 교수

관심분야 : 영상처리, 통신시스템, 로봇비전, 임베디드 시스템