

웹 서버를 위한 Embedded Linux 시스템 설계 및 구현에 관한 연구

최병윤* · 고성찬*

*안동대학교

Embedded Linux System Design for web server and study of embodiment

Byoung-yun Choi* · Sung-chan Ko

* Andong National University

E-mail : goust113@hotmail.com

요 약

대부분의 기존 임베디드 웹 서버 시스템은 어떤 특정 기기를 제어하기 위하여 고안되고 있다.

본 논문에서는 StrongArm 계열의 SA1110 chip을 채택하여 H/W 보드를 설계하고 최적의 웹 서버를 만들었다. 최근 널리 사용되고 있는 SA1110 H/W 보드 설계 중 Ethernet Controller인 CS8900A 설계에 관한 세부 사항들과 Ethernet 관련 Kernel patch에 대한 내용을 다루었다. 그리고 CS8900A에 대한 Network device driver 모듈에 대하여 자세히 언급하며, 제작된 보드를 사용하여 웹 서버 프로그램을 작성하여 MS 익스플로러 6.0환경에서 실험하였다.

I. 서 론

정보통신의 발전과 인터넷 서비스의 다양화로 인하여 사회 전반에 걸쳐 많은 변화가 일어나고 있다. 그 일환으로 임베디드 시스템인 전자 시스템이 나왔다.

임베디드 시스템(Embedded System)이란 미리 정해진 특정한 기능들을 수행하기 위하여 컴퓨터의 하드웨어와 소프트웨어가 결합된 고기능의 전자 제어 시스템을 말한다. 즉, 우리 실생활에서 사용되고 있는 각종 전자 기기, 가전 제품, 제어 장치는 단순히 회로만으로 구성된 것이 아니라 마이크로프로세서가 내장되어 있고, 그 마이크로프로세서를 구동하여 특정한 기능을 수행하도록 프로그램이 입력되어 있는 내장 시스템을 가진다. 일상 생활에서 사용되고 있는 TV, 냉장고, 세탁기, 전자 레인지 제품뿐만 아니라 PDA, 핸드폰, 사이버 아파트의 홈 관리 시스템, 홈 네트워크 게이트웨이 시스템, 교통 관리 시스템, 현금 지급기(ATM), 항공 관제 시스템, 우주선 제어 장치, 군사용 제어 장치 등 셀 수 없는 많은 기술들이 내부적으로 임베디드 시스템 기술이 적용되어 우리 생활과 아주 밀접하게 관련되어 있다.

이처럼 가정용 혹은 산업용 기기를 인터넷에 연결하여 다양한 서비스를 제공하고자 하는 요구가

급증하고 있으나 대부분의 기기는 RS-232C나 적외선 리모컨 같이 매우 단순한 통신 프로토콜을 내장하거나 통신 기능이 전혀 없다. 따라서 이러한 기기를 인터넷에 연결하기 위해서는 일종의 게이트웨이 기능을 할 수 있는 특별한 연결 장치가 요구된다.

일반적으로 웹 서버를 이용하여 기기를 감시 및 제어 할 경우 여러 가지 방법이 있으나 임베디드 웹 서버를 들 수 있다.

본 논문에서는 SA1110 CPU의 고속 프로세서를 이용한 임베디드 웹 서버를 범용 웹 서버와는 달리 저렴하고 가정용 혹은 산업용으로 쓰일 수 있는 모델을 제시하고, 확장성을 제공할 수 있는 초소형 펌웨어 설계와 웹서버 기능에 최적화된 S/W 모듈을 설계 및 네트워크 디바이스 드라이버인 CS8900A에 대하여 다루었다.

II. 펌웨어 설계

본 논문에서는 저렴한 가격의 초소형 웹 서버를 구현하기 위하여 그림 1과 같이 Intel StrongArm 인 SA1110을 사용하였다. 이 CPU는 32비트 RISC (Reduced Instruction Set computer) 구조로 명령어가 간단하고, 동작 속도가 2.1MIPS인 206MHz로

빠르며, 낮은 소비전력(400mW-1.75V)을 가지고 있으며, 제공되는 주변 장치의 집적도 또한 좋으며, 성능이 우수하여 PDA 및 고성능 핸드폰 단말기 등을 위한 플랫폼에 많이 사용되고 있다.

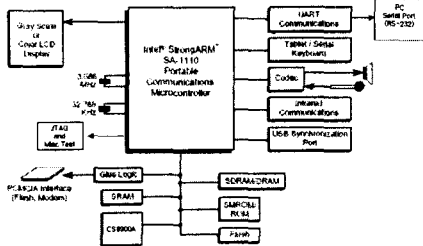


그림 1. 임베디드 웹 서버 펌웨어 구성도

StrongArm코어는 주변 장치 컨트롤러는 Interrupt Controller, Serial Controller, LCD Controller, Memory Controller가 있고, 일반적인 입출력으로 GPIO(General Purpose I/O)가 있다. 클럭으로는 RTC(Real-Time Clock)와 Interval Clock이 있다. 또한 무선 LAN을 위한 PCMCIA Controller가 있으며, USB(Universal Serial Bus) Controller, 115Kbps~4Mbps를 지원하는 적외선 IrDA Controller가 있다. Motorola SPI, national Microwire, TI Synchronous Serial, Phillips UCB1110/UCB1200을 지원하는 CODEC 인터페이스도 지원한다.

임베디드 웹 서버를 인터넷을 연결하기 위해 사용될 이더넷 컨트롤러는 현재 가장 많이 쓰이는 Cyrus Logic에서 개발된 10Mbps를 지원하는 CS8900A 칩이다. 이 칩은 IEEE 802.3기준을 따르는 이더넷 컨트롤러로서, CSMA/CD(Carrier Sense Multiple Access/ Collision Detcet)방식을 쓴다. CS8900A는 전이중 통신(Full-duplex)를 지원하며, 칩 자체에 Tx와 Rx를 위한 buffer를 가지고 있다. 네트워크 인터페이스로는 10BASE-T와 10BASE2, 10BASE5, 10BASE-F를 지원하기 위한 AUI(Attachment Unit Interface)를 가지고 있으며, Tx시에는 collision이 있을 때, 자동적인 재전송 및 Padding과 CRC 생성을 해 준다. Rx시에는 CPU의 overhead를 줄이기 위해서 Stream Transfer를 제공한다.

임베디드 웹 서버로 RJ-45 소켓에 UTP 랜 케이블을 연결하여 인터넷에 접속되는 구조를 가지고 있으며, 실생활에 쉽게 적용 가능한 임베디드 서버로 개발하여 다양한 인터페이스를 고려하였다.

III. S/W 설계

S/W는 웹 서버의 기능을 충실하고 안정적으로 수행할 수 있도록 불필요한 기능은 배제하고, 각각의 모듈을 최적화 하여 최소의 코드 사이즈로 필요

로 하는 기능을 수행 가능하도록 설계하였다. 그림 2에서는 웹 서버에서 HTTP가 사용자의 웹 브라우저로부터 요청된 웹 페이지 데이터를 전송하는 기능을 담당하는 모듈로서 개략적인 제어 흐름을 나타낸 것이다. 임베디드 웹 서버는 특성상 클라이언트로부터 전송되는 정보의 양이 비교적 작아서 HTTP GET method를 사용하는 것이 메모리 관리와 데이터 저리에 효율적이다. 따라서 본 논문에서는 HTTP GET method만을 지원하도록 설계한다.

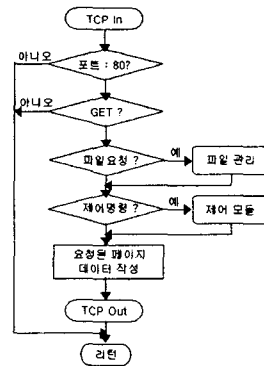


그림 2. HTTP 모듈 흐름도

HTTP 모듈은 클라이언트 웹브라우저의 HTTP GET 요청에 대해 HTML 문서, 텍스트, 이미지, 자바 스크립트, css 등을 전송함으로써 실제 웹 서버와 같은 응답을 한다. 클라이언트의 직접적인 HTTP GET 요청에 대한 웹 페이지나 이미지, 자바 스크립트 등의 반환은 임베디드 웹 서버의 기억 공간에 저장된 데이터를 이용하여 실제 파일시스템 처럼 처리한다. 이러한 동작을 위해 파일 관리 모듈을 이용한다.

IV. 네트워크 디바이스 드라이버 모듈

웹 서버를 인터넷과 연결을 하기 위해서는 SA1110 과 CS8900A를 연결해 주어야 하며, CS8900A의 블록도는 그림 3과 같다. 외부의 serial EEPROM에 의해 configuration이 수행될 수 있으며, 10Base-T 또는 10Base2, 10Base5, 10Base-F접속을 수행할 수 있다.

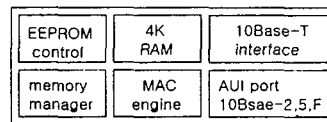


그림 3. CS8900A 블록도

내부에 존재하는 4Kbyte RAM(PacketPage) 메모리는 내부 레지스터, 송신데이터 버퍼, 수신데이터 버퍼로 사용된다.

PacketPage를 액세스하는 방식은 3가지(me-

mory mode, I/O mode, DMA mode)가 있다. CS8900A에서 I/O 모드에서 I/O base address를 할당하고 아래 표 1에 주어진 offset을 가지고 CS8900A I/O 모드 매핑에 따라 이더넷 컨트롤러와 데이터를 송수신 행하는 방식이다. 이는 AEN, IOR, LOW 핀을 가지고 사용하며 AEN이 low, IOR과 IOW 또한 low 일 때 I/O read/write 동작을 한다.

표 1. I/O 모드 매핑

OFFSET	TYPE	Description
000h	R/W	송수신 데이터(16bit)
002h	R/W	송수신 데이터(16bit) : 32비트 동작시 사용
004h	W	TxCMD
006h	W	TxLengh
008h	R	ISQ(interrupt status queue)
00Ah	R/W	PacketPage pointer
00Ch	R/W	PacketPage data(16bit)
00Eh	R/W	PacketPage data(16bit) 32비트 동작시 사용

TxCMD는 프레임 송신을 알리고, TxLength는 송신할 프레임의 길이, ISQ는 인터럽트 상태 레지스터, PackPage Ptr은 CS8900A의 내부 레지스터 액세스를 위한 포인터이다.

SA1110과 CS8900A와의 연결은 그림 4와 같이 연결한다. nCS1은 physical address 0x08000000에 mapping되어 있으며, A[0:25]는 0x300h를 base address로 한다.

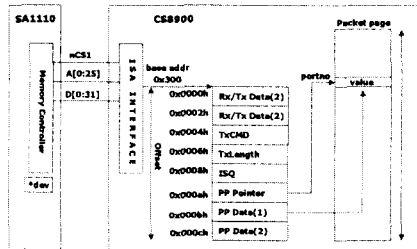


그림 4. 16비트 I/O 포트

그러므로 CS8900의 SA1110에서의 address는 0x08000300이다.

이것은 BUS로 나가는 물리적인 주소이고, MMU(Memory Management Unit : 기억 관리 장치)를 사용할 경우는 0xd8000300에 매핑된다.

리눅스에서는 이러한 정보를 디바이스 구조체 포인터의 base_addr에서 포인팅하도록 되어있다.

CS8900 I/O Mode Mapping를 사용하자면 Packet Page Pointer register를 통해서 접근하면 된다. 파일 위치는 ~/driver/net/cs890x.c에 있으며, readreg()와 writereg() 함수로 접근한다. readreg() 함수는 net_device 구조체와 쓰게 될 포트 번호(0x40 : EEPROM command)를 파라미터 값(0x20)으로 받는다. 먼저 기본 주소(device I/O address : 0x300)에 ADD_PORT(=0x000A)를 더한

후(Packet Page Pointer Register 선택), 이곳에 해당 레지스터의 오프셋을 넣은 후, 기본 주소에 데이터 포트의 오프셋(= DATA_PORT : 0x000C)을 더한 주소를 읽어들이어서 복귀 값으로 전달한다.

writereg()함수도 같은 일을 하는데, net_device 구조체와 포트 번호 및 쓸 값을 변수로 넘겨받아서, 기본 주소에 ADD_PORT를 더한 주소에 포트 번호를 쓰고, 다시 기본 주소에 데이터 포트의 주소를 더한 곳에 값을 쓴다.

포트에 값을 바로 읽고 쓰기 위해서는 readword() 함수와 writeword() 함수를 사용한다. 각각은 다시 하위의 루틴으로 raw_readw()와 raw_writew()를 사용한다.

모듈의 적재와 해제, 네트워크 초기화, 네트워크 열기와 닫기, 상태 역시 cs890x.c 파일 안에 있다.

먼저 모듈 적재와 해제는 적재 시에는 register_netdev() 함수를 호출하고 해제하기 위해서는 unregister_netdev()를 호출하면 된다.

cs890x_probe() 함수는 실제적으로 시스템에서 CS8900A를 사용한 이더넷 컨트롤러가 있는지를 찾는다. 하지만 실제체적으로 칩을 검색하는 함수는 cs890x_probe1()이 맡고 있으며, 초기화에서만 사용하고 더 이상 사용을 하지 않는다. 넘겨받은 net_device 구조체가 기본 I/O 주소를 가지는 ioaddr이며, 먼저 network adapter의 private 데이터 저장공간으로 사용할 부분을 커널 메모리에서 할당을 받아 이 공간을 lp로 가리키고, 전부 0으로 설정한다(memset()). 그 후 칩의 선택을 결정하기 위해서 readreg() 함수를 호출한다. 그리고 EEPROM이 존재하면 MAC 주소를 EEPROM에서 읽어오도록 한다. 그 후 인터페이스로 사용하는 함수들을 정의해준다.

get_eeprom_data() 함수는 EEPROM에서 정해진 만큼의 데이터를 읽어오는 일을 한다. get_eeprom_data() 함수와 get_eeprom_cksum() 함수는 EEPROM 으로부터 데이터를 읽고, 체크섬을 비교하는 함수이다. wait_eeprom_ready() 함수를 호출해서 eeprom이 ready인지를 확인하고, PP_EECMD 레지스터에 해당 offset과 EEPROM_READ_CMD를 OR시켜서 쓴다. 다시 EEPROM이 ready상태인지를 확인한 후, PP_EEData 레지스터를 읽어서 buffer에 저장한다.

get_eeprom_cksum()은 읽은 EEPROM의 데이터를 다 더해서 0xFFFF와 AND시켜서 이 값이 0을 가진다면 0을 돌려주고, 그렇지 않다면 -1을 돌려준다. 즉, 읽은 데이터의 합은 0이 되어야 맞다.

net_open() 함수는 네트워크 디바이스의 초기화(initialization) 이후에 실제로 네트워크 interface를 사용하기 위해서 가장 먼저 호출되는 함수이다. 따라서, 앞에서 초기화 과정에서 해주지 않았던 초기화와 관련된 설정과, 버퍼 등의 관리를 위해서 해주어야 할 일들 및 interrupt를 사용 가능한 상태로 만들어 주어야 한다. buffer관리를 위해서 사용하는 interrupt 번호가 무엇인지를 알려주고(write_irq()), 해당 인터럽트를 요청한다(request_irq()). 그 후 RJ45를 사용하는 Full Duplex, Half Duplex 및 AUI와 BNC에 대해서 찾았다는 것을 나타낸다. 모든 설정이 끝났으므로,

Tx와 Rx를 활성화 시켜준다.

네트워크 인터페이스에 대한 사용을 마치기 위해서 커널은 close() 함수를 호출할 것이다. CS8900에서는 net_close() 함수를 정의하고 있다.

먼저, 더 이상 이 network 인터페이스에 대한 패킷의 큐를 사용하지 않는다는 것을 알려주기 위해서 netif_stop_queue()를 호출한다. 그 후 PP_RxCFG와 PP_TxCFG, PP_BufCFG, PP_BusCTL에 각각 0을 쓴다. 할당받은 interrupt를 해제하기 위해서 free_irq() 함수를 호출한다.

net_send_packet() 함수는 커널이 패킷을 보내고 싶을 때 호출된다. 중요한 것은 넘겨받는 파라미터로 socket buffer를 받는다는 것이다. 또한 해당하는 net_device 구조체를 전달받는다.

net_rx() 함수는 네트워크 디바이스 드라이버의 받기(receive)와 관련된 연산은 항상 interrupt에 관련된 부분이다. 즉, 보내기와 관련된 것은 커널과 interface가 정해져 있지만, 받는 것은 일반적인 디바이스 드라이버에 대한 read 연산과 같은 관련성이 없다. 단지 인터럽트가 발생했을 때, 해당 frame을 읽어서, 상위의 프로토콜 layer에 알려주기만 할뿐이다. 따라서, 인터럽트의 발생 시에 이 인터럽트가 Rx 인터럽트인지 확인한 후, 이를 처리해준다.

net_get_stats() 함수는 네트워크 디바이스의 통계 정보를 얻기 위해서 사용된다.

IV. 구현 및 실험

본 논문에서 제시한 임베디드 웹 서버 모델의 활용 가능성을 평가하기 위하여 다음과 같은 환경에서 S/W 모듈을 구현하여 기본적인 기능들을 실험하였다.

◇ 구현 환경

임베디드 웹 서버의 S/W 모듈은 GNU 공개 파일인 arm linux gcc compiler를 사용하였다. 커널은 2.4.5이며, ARM patch인 rmk7, SA1110 patch인 np2을 적용시키고, 마지막으로 웹 서버에 맞게 patch를 적용시켜서 사용하였다. 디버깅과 소프트웨어 적재를 위하여 LART(www.lart.tudelft.nl)에서 제공하는 blob를 사용하여 Flash Rom에 적재하였다. blob의 기능은 CPU clock, Memory timing, interrupt, UART등을 초기화하며, 리눅스 부팅과 각 종 이미지의 다운로드를 담당하게 하였다.

설계된 임베디드 웹 서버의 실험 환경으로는 제작된 펌웨어와 서버에 내장될 소프트웨어 프로그램으로 웹 페이지를 구성하기 위한 HTML과 Flash로 작성한 이미지, 자바 스크립트, css 등으로 구성되어 있으며, 웹 서버로는 boa(www.boa.org)에서 제공하는 boa 웹 서버를 사용하였다.

◇ 실험

본 논문에서 설계 구현된 임베디드 웹 서버의 활용 가능성을 검증하기 위해 기본적인 웹 서버의 정상 동작을 확인하기 위한 기본페이지 출력 실험은 다음과 같다.

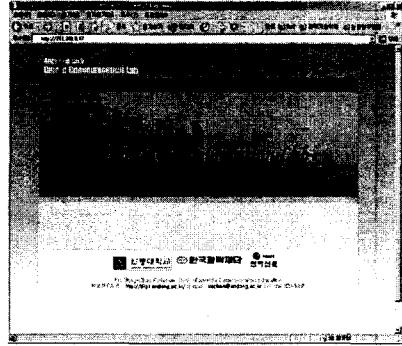


그림 5. 웹 서버 홈페이지 출력 결과

실험에서는 MS 익스플로러 6.0으로 임베디드 웹 서버에 설정한 IP 접속시 Flash Rom에 저장되어 있는 기본 웹 페이지(index.html과 main_intro.swf, 그 외 이미지 파일)가 정상적으로 전송되어 그림 5와 같이 웹 브라우저에 출력되었다. 이것으로 임베디드 웹 서버가 정상적으로 동작하는 것을 확인하였다.

V. 결론

본 논문에서는 임베디드 웹 서버의 모델을 제시하고 StrongARM 계열인 SA1110 프로세서를 이용하여, 인터넷과 연결을 위하여 CS8900A 이더넷 컨트롤러를 사용하여 시험용 보드를 제작하여 활용 가능성을 검토하였다.

제시된 임베디드 웹 서버는 다양한 언어를 사용할 수 있으며, 기존 웹 서버 제품의 비싼 가격 대신 저렴하고 초소형의 제품으로 대체할 수 있다.