

# 임베디드 시스템 기반 동적호스트 서버 구현

김용호<sup>\*</sup> · 박종현<sup>\*</sup> · 오근탁<sup>\*</sup> · 김형균<sup>\*\*</sup> · 최광미<sup>\*\*</sup>

조선대학교<sup>\*</sup> · 동강대학<sup>\*\*</sup>

## Dynamic Host Server Implementation of Based Embedded System

Yong-ho Kim<sup>\*</sup> · Jong-heon Park<sup>\*</sup> · Keun-Tack Oh<sup>\*</sup> · Hyeong-Gyun Kim<sup>\*\*</sup> · Gwang-mi Choi<sup>\*\*</sup>

Chosun University<sup>\*</sup> · Dongkang College<sup>\*\*</sup>

E-mail : kimyh@mail.chosun-c.ac.kr

### 요 약

본 논문에서는 임베디드 시스템 기반으로 한 동적 호스트 구성 프로토콜 서버를 구현하는데 목적을 둔다. 이를 위해 개발 보드로 Intel Strong ARM SA-1110 프로세서를 탑재한 ez board-M01에 임베디드 리눅스를 포팅하고 네트워크 기능을 위하여 이더넷을 기반으로 한 기본적인 네트워크를 구축하였다. 이를 통해 임베디드 보드에서 동적으로 네트워크 정보를 할당하여 윈도우 클라이언트 호스트와 리눅스 클라이언트 호스트가 동적으로 네트워크 정보가 설정되는 임베디드 DHCP 서버를 제안하고자 한다.

### ABSTRACT

The purpose of this study is to realize DHCP server based on embedded system. To achieve this, embedded Linux was ported in ez Bord-M01 mounted with Intel Strong ARM SA1110 processor, and ethernet-based network was constructed for network function. In this way, this study suggests embedded DHCP server where Window and Linux client hosts are dynamically configurated as network information by dynamically assigning network information in embedded board.

### 키워드

Embedded System, 동적호스팅

## 1. 서 론

인터넷이란 매체를 통하여 전 세계가 하나로 연결된 지금, 데이터 통신은 컴퓨터의 필수 구성 요소가 되었다. 인터넷이 일상화되면서 일반 사용자가 네트워크에 쉽게 접근할 수 있도록 여러 기술들이 끊임없이 개발되었고, 그 결과 많은 프로토콜이 표준화 되었는데 그 중 하나가 DHCP이다.

DHCP(Dynamic Host Configuration Protocol)는 IP 주소 및 TCP/IP 프로토콜 기본 설정을 개별 호스트에 동적으로 할당하는 프로토콜로써, 일반 사용자들이 네트워크 설정 방법을 몰라도 TCP/IP를 이용한 호스트 간 통신을 가능하게 해준다. 네트워크 규모가 크고 복잡할수록 DHCP의 이런 기능은 거대 네트워크 내의 네트워크 설정

을 쉽게하여 네트워크 관리자와 일반사용자를 지원하는 역할을 한다.

DHCP는 시스템 부팅시, RARP를 사용하지 않고도 호스트 자신의 IP주소를 결정할 수 있도록 지원하며 IP 주소 외에도 게이트웨이, 서브넷 마스크, 네임 서버 등의 네트워크 정보를 동적으로 할당해 준다.

네트워크 장비의 대부분은 특정 기능을 수행하도록 설계되어진 임베디드 시스템이다. 임베디드 시스템이란 미리 정해진 특정 기능을 수행하기 위해 컴퓨터의 하드웨어와 소프트웨어가 조합된 전자 제어 시스템이다. 일반 범용의 다양한 기능을 수행하는 컴퓨터가 아닌 시스템 고유의 목적에 맞추어 정형화된 기능만을 수행하도록 설계된 컴퓨터 시스템을 말한다. 초창기 임베디드 시스템

은 특정 기능에 최적화, 초소형, 저전력을 겸비한 단순한 기능을 수행하는 정도였으나, 점차 시스템의 복잡성과 다양성을 만족시키기 위한 시스템으로 발전하고 있다.

본 논문에서는 임베디드 시스템 기반으로 한 동적 호스트 구성 프로토콜 서버를 구현하는데 목적을 둔다. 이를 위해 개발 보드로 Intel Strong ARM SA-1110 프로세서를 탑재한 ez board-M01에 임베디드 리눅스를 포팅하고 네트워크 기능을 위하여 이더넷을 기반으로 한 기본적인 네트워크를 구축하였다. 이를 통해 임베디드 보드에서 동적으로 네트워크 정보를 할당하여 윈도우 클라이언트 호스트와 리눅스 클라이언트 호스트가 동적으로 네트워크 정보가 설정되는 임베디드 DHCP 서버를 제안하고자 한다.

## II. 임베디드 리눅스 시스템

### 2.1 임베디드 리눅스

Linux는 초기 PC나 서버급 시스템에 포팅되어 사용되다 최근 임베디드 시스템으로 그 관심사가 옮겨진 상태다. 이미 국내외에서는 임베디드 시스템에 리눅스가 포팅되어 많은 제품들이 출시되었다. 임베디드 OS 분야에서 임베디드 리눅스의 비중은 점점 커져가고 있다. 기존의 임베디드 OS와 비교해 볼 때 상용 OS보다는 임베디드 시스템이 가지는 실시간적인 요소를 충족시키지 못하는 것은 사실이다. 또한 Windows CE 보다는 개발환경이 좋은 편도 아니다. 그리고 본래 PC 기반으로 만들어 졌기 때문에 메모리가 열악한 임베디드용으로 쓰기에 커널이 너무 크다. 이런 불리한 점을 가진 임베디드 리눅스가 그 비중이 커지고 사람들이 많이 다루는 이유는 오픈 소스에 라이선스 비용이 없다는 것이 임베디드 리눅스가 가지는 큰 장점이기 때문이다. 본 논문에서는 이러한 장점으로 임베디드 리눅스를 개발 보드의 OS로 선정 하였다.

### 2.2 Windows CE

과거의 임베디드 시스템은 사용자 인터페이스가 아예 없거나 조약하기 이를 데 없었다. 하지만 시간이 흐르면서 기술 발전에 힘입어 임베디드 시스템도 화려한 사용자 인터페이스를 장착하기 시작했으며, 특히 디지털 정보기기가 출현하면서 이에 대한 관심은 더욱더 커졌다. 하지만 이런 요구를 맞추다보니 하드웨어의 사양이 매우 높아져서, 가격 경쟁력에서 다른 모바일 제품에 밀리고 있는 실정이다. 네트워크 기능을 보면 적외선 통신, 데스크탑과의 오토싱크, 그리고, PCS나 셀룰러 폰

을 이용한 웹 브라우징 능력을 갖고 있다. 특히, 차세대 가전 제품에 Windows CE를 이용하여 하나의 윈도우 호환 환경에서 홈 오토메이션, 카 내 비게이션 등을 고려함으로써 하드웨어 업체들의 강력한 지원을 받았다. 따라서, 개발 환경은 다른 임베디드 OS에 비해 훨씬 편리하다.

### 2.3 임베디드 자바

1990년에 Sun Microsystem에서 Green 프로젝트로 개발된 Oak는 전자제품에서 사용할 소프트웨어로 개발되었다. 이 소프트웨어는 다른 시스템에 대해서 이식성이 매우 높았으나 전자제품과 같은 임베디드 시스템에서 쓰이지 못하고, 인터넷 프로그래밍에 쓰일 수 있다는 가능성을 발견해 Oak를 인터넷 프로그래밍이 가능하게 개발하여 만든 것이 Java이다. Java가 임베디드 시스템에서 사용될 수 있도록 한 대표적인 것이 Sun의 임베디드 Java와 퍼스널 Java이다.

Java의 가장 큰 장점은 어떠한 시스템에도 Java API와 JVM(Java Virtual Machine)만 있으면 자바 코드가 이식이 가능하다. Java API와 JVM이 내장하고 있는 임베디드 Java와 퍼스널 Java는 기존의 상용 RTOS와 연계되어 사용된다.

### 2.4 pSOS

ISI에서 1980년대에 개발한 pSOSSystem은 우리나라의 여러 업체가 채택해서 사용하고 있는 RTOS로 삼성전자가 pSOS+ 개발에 참여해 라이선스를 갖고 있다. 삼성전자의 휴대폰에 사용되어 왔으며, 각종 통신장비와 네트워크 장비에서 사용되고 있다. pSOSSystem은 커널을 중심으로 해서 여러 개의 소프트웨어 컴포넌트들로 구성되어 있다. 이들 소프트웨어 컴포넌트들은 각각의 독립적인 모듈로 되어 있으며 통합개발환경 툴로 pRISM+을 제공하고 있다.

pSOSSystem은 멀티태스킹 RTOS로 각 태스크들은 우선순위를 가지고 있어 우선순위가 높은 태스크들의 작업 수행이 먼저 이루어진다. 따라서 선점형 스케줄링 방식을 따른다고 볼 수 있다. 만일 각 태스크들이 같은 우선순위를 가진다면 스케줄링 방식은 라운드로빈(Round-robin) 방식으로 바뀌게 된다. 태스크의 수는 총 256개이다. 그리고 태스크 관리, 세마포어(semaphore), 메시지 큐, 시간 관리 및 타이머, 이벤트 및 비동기 시그널, 에러 처리, 동적인 메모리 저장관리, 다른 태스크들로 부터의 코드나 데이터 보호 등의 서비스를 지원한다. 여러 개의 다른 실행 모드를 가지고 있는 CPU를 위해서 사용자 모드와 슈퍼바이저 모드를 제공하고 있다.

### III. 구현 및 실험

#### 3.1 시스템 구성도

임베디드 시스템을 구현하기 위하여 타겟 보드로 Intel Strong ARM SA-1110 프로세스 기반의 이더넷 보드를 선정하고 호스트 PC에 레드햇 리눅스 9.0을 설치하여 개발 PC를 구성하였다. 네트워크 환경이 설정되지 않은 리눅스 시스템과 윈도우 시스템을 이더넷 환경에 연결시켜 그림1과 같은 네트워크 환경을 구축하였다

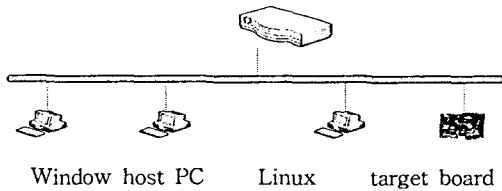


그림 1. 시스템 구성도

본 논문에서는 호스트 PC에서 크로스 컴파일 환경을 구축하여 dhcp 소스파일을 컴파일하여 설치한 후, 타겟 보드에 시리얼 통신을 이용하여 다운로드 한 후 타겟 보드에 동적 호스트 구성 프로토콜 서버를 구현 하였다.

#### 3.2 크로스 컴파일 환경 구축

임베디드 시스템에서의 개발은 타겟 보드 상에서 이루어질 수 없다. 특히 운영체제 포팅, 커널 컴파일 등을 처리하기 위해서는 보드에 맞는 환경을 구축하고, 각종 장치 드라이버를 적재하는데 작업을 할 수 있는 환경이 필요하다.

#### 3.3 시리얼 통신 환경구축

크로스 개발 환경이 갖춰진 호스트 PC에서 타겟 보드를 조작하고 개발된 응용프로그램을 다운로드하기 위하여 호스트 PC에 minicom 프로그램을 설치하고 그림2와 같이 환경 설정을 하였다.

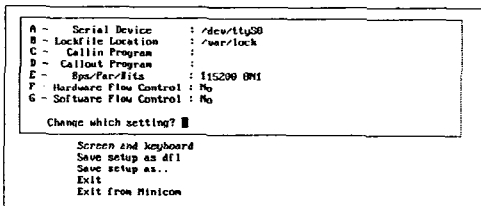


그림 2. minicom 환경 설정

#### 3.3 동적 호스트 구성 프로토콜 서버 구현

동적 호스트 구성 프로토콜 서버를 구현하기 위하여 dhcp 소스 프로그램을 그림3와 같이 호스트 PC에 설치하였다.

```

[aipjh@ssl embedded]$ tar xvfz dhcp.tar.gz
[aipjh@ssl embedded]$ cd dhcp
[aipjh@ssl dhcp]$ ./configure
[aipjh@ssl dhcp]$ make all
[aipjh@ssl network]$ su - root
Password: *****
[root@ssl /]# cd /home/aipjh/embedded/dhcp
[root@ssl dhcp]# make install
    
```

그림 3. dhcp 서버 설치

설치를 끝낸 후 dhcp 클라이언트 네트워크 정보를 동적으로 할당하기 위하여 그림 4와 같이 dhcp 서버의 환경을 설정 하고, 서버가 할당한 시스템의 MAC 주소와 IP 주소를 기록하기 위한 로그 파일을 그림 5와 같이 생성하였다.

```

[aipjh@ssl dhcp]$ cat >> /etc/dhcp.conf
subnet 210.182.203.0 netmask 255.255.255.0;
option routers                210.182.203.254;
option subnet-mask            255.255.255.0;
option domain-name            "kjcci.or.kr";
option domain-name-servers   168.126.63.1;
range dynamic-bootp           210.182.203.205
210.182.203.210;
default-lease-time 21600;
max-lease-time 43200;
^D
[aipjh@ssl dhcp]$
    
```

그림 4.. dhcp 서버 환경 설정 파일 생성

```

[aipjh@ssl dhcp]$ touch
/var/lib/dhcp/dhcp.leases
[aipjh@ssl dhcp]$
    
```

그림 5. dhcp 서버 로그 파일 생성

실험결과를 확인하기 위하여 윈도우 시스템과 리눅스 시스템 각각의 호스트를 부팅하고 네트워크 연결 상태와 네트워크 정보를 확인하기 위하여 ping, ipconfig, ifconfig 명령을 이용하여 표 1과 같은 임베디드 dhcp 서버 구현

전·후 결과 값을 얻었다.

표 1. 임베디드 dhcp 서버 구현 전후 비교

항목	구현	임베디드 dhcp 구현 전	임베디드 dhcp 구현 후
윈도 dhcp Client	Ping 결과	<pre>                     ping 210.182.203.201                     Destination host unreachable                     Destination host unreachable                     Packets: Sent = 4 Received = 0 Lost = 4 (100% loss)                 </pre>	<pre>                     ping 210.182.203.201                     Reply from 210.182.203.201: bytes=32 time=4ms TTL=62                     .                     Reply from 210.182.203.201: bytes=32 time=4ms TTL=62                     Packets: Sent = 4 Received = 4 Lost = 0(0% loss)                 </pre>
	ipconfig 결과	<pre>                     ipconfig                     IP Address . . . . . :                     Subnet Mask . . . . . :                     Default Gateway . . . . . :                 </pre>	<pre>                     ipconfig                     IP Address . . . . . : 210.182.203.205                     Subnet Mask . . . . . : 255.255.255.0                     Default Gateway . . . . . : 210.182.203.204                 </pre>
Linux dhcp Client	Ping 결과	<pre>                     ping 210.182.203.201                     Destination host unreachable                     Destination host unreachable                     11 packets transmitted: 0 packets received, 49% success                     100% packet loss                 </pre>	<pre>                     ping 210.182.203.201                     Reply from 210.182.203.201: bytes=32 time=4ms TTL=62                     .                     Reply from 210.182.203.201: bytes=32 time=4ms TTL=62                     Packets: Sent = 4 Received = 4 Lost = 0(0% loss)                 </pre>
	ifconfig 결과	<pre>                     ifconfig eth0                     eth0: flags=4096&lt;BROADCAST,MULTICAST&gt;                     HWaddr 00:00:00:00:00:00                     Bcast 255.255.255.0                     Mask 255.255.255.0                 </pre>	<pre>                     ifconfig eth0                     eth0: flags=4096&lt;BROADCAST,MULTICAST&gt;                     HWaddr 00:00:00:00:00:00                     Bcast 255.255.255.0                     Mask 255.255.255.0                 </pre>

윈도우와 리눅스 시스템 모두, dhcp 서버 구현 전에는 IP, Subnet Mask, Default Gateway가 할당되지 않았으나 서버 구현 후 210.182.203.205와 210.182.03.206번의 IP가 리눅스와 윈도우 시스템 각각에 할당된 걸 확인할 수 있다. 또한 ping 테스트 결과 모든 패킷을 호스트 PC에 송수신하는 걸 실험결과로 확인 하였다.

#### IV. 결 론

네트워크 상에 새로운 호스트를 접속하고자 할 때 관리자는 그 호스트에 직접 IP 주소 및 네트워크 정보를 설정하여야 하지만 휴대형 컴퓨터와 같이 일시적으로 IP 주소를 필요로 하는 경우에는 비 효율적이다. DHCP는 IP 주소의 효율적인 관리와 네트워크 호스트들을 중앙에서 관리할 수 있도록 등장한 프로토콜로 관리자의 네트워크 관리 부담을 덜어준다. 그러나 기존의 DHCP 서버 장비는 고가라서 네트워크 규모가 적은 곳에서 사용하기에는 비효율적이다.

본 논문에서는 Intel Strong ARM SA-1110 프로세서 기반의 ez board-M01에 공개 운영체제인 리눅스를 포팅하고 이더넷을 통하여 동적 호스트 구성 프로토콜 서버를 구현 하였다. 이를 위하여 리눅스 기반의 호스트 PC와 타겟 보드를 시리얼 통신으로 연결하고 네트워크 정보가 설정되지 않은 윈도우 클라이언트와 리눅스 클라이언트를 이더넷 환경에 연결하여 동적으로 네트워크 정보가 할당되도록 실험 환경을 구축하였다.

본 연구에서 제시하는 임베디드 시스템을 이용한 동적 호스트 구성 프로토콜 서버는 DHCP 본래의 기능을 수행하면서도 가격은 저렴하고, 크기

를 축소한 동적 호스트 구성 프로토콜 서버이다. 그러나 기술적 측면에서 본 논문에서 구축된 임베디드 시스템을 거대한 네트워크 내부에서 더욱 성능이 우수하고 안정적인 네트워크 정보를 할당하기 위해서는 더욱 더 안정적인 임베디드 보드의 개발이 필요하다. 또한 무선 인터넷 환경하에서 본 연구에서 구현된 임베디드 동적 호스트 구성 프로토콜 서버에 대한 지속적인 연구가 필요 하리라 예상된다.

#### 참고문헌

- [1] <http://www.embedded.com>, "Linux as an Embedded Operation System".
- [2] <http://www.emlinux.com>, "Embedded Linux in a Commercial Product".
- [3] James Y. Wilsons, Aspi Havewala, "Building Powerful Platforms with Windows CE", Addison-Wesley Publishers, 2001
- [4] Real Time Magazine, "Comparison between VxWorks/x86 5.3.1, QNX 4.25 and pSOSytem/x86 2.2.6", April 1999.
- [5] Moshe Bar, "Linux File System", McGraw-Hill, 2001
- [6] Stevens, W. Richard, "TCP/IP Illustrated. Vol. 2, The Implementation", Addison Wesley, 1994.