

# ARM프로세서를 이용한 RS232C와 TCP/IP 접속장치의 구현

이영준, 한경호  
 단국대학교 전기공학과

## Implementation of RS232C and TCP/IP Connection Device Using ARM Processor

Youngjun Lee, Kyongho Han  
 Dept. of Electrical Engineering, Dankook University

### ABSTRACT

In this paper, the connection device of RS232C and TCP/IP implementation using ARM processor and LINUX is proposed. Data interaction flash memory the multiple serial ports are transferred to ARM processor and the data are processed and formed into data packet for transfer via internet protocol. Packet flash memory internet is decoded to extract the serial port data. The serial ports supports RS232C asynchronous protocol communication and control program is developed in GNU-C and installed in the on-board memory for packet conversion and control. The research result can be applied to terminal server, printer server and multiple serial ports equipments.

### 1. 서 론

본 연구에서는 LINUX를 기반으로 한 ARM 프로세서 보드의 TCP/IP 프로토콜을 이용하여 인터넷망에 접속하기 위한 인터페이스를 구현하는 것을 연구 목적으로 한다. 이를 위하여 ARM 프로세서 보드에 Embedded LINUX OS를 실장하고 C 언어에 의한 SOCKET 접속 제어 프로그램을 구현하여 실장하고 PC등과 LAN 망을 통하여 접속하여 원격 프로세서의 시리얼 포트를 LAN을 통하여 데이터 전송하도록 한다.

또한 ARM 프로세서 보드의 시리얼 포트의 수를 확장하여 다수의 시리얼 포트를 LAN 망에 접속할 수 있도록 하여 터미널 서버의 기초 기능을 구현하고자 한다.

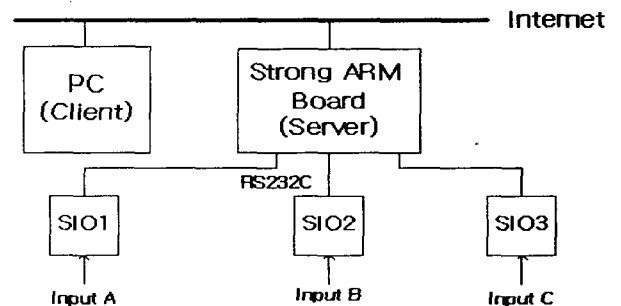


그림 1 터미널 서버 구성도

## 2. ARM프로세서를 기반으로 LINUX Board에서의 TCP/IP통신

### 2.1 개발환경

사용한 하드웨어 개발환경으로 HyBus 보드를 사용하였으며 보드의 주요 구성부품은 Intel Strong ARM SA1110 Processor (32Bit206MHz), SDRAM (32Mbyte), 플래쉬 메모리(16Mbyte)로 구성되어 있다. 소프트웨어는 GNU-C 컴파일러를 사용하였다.

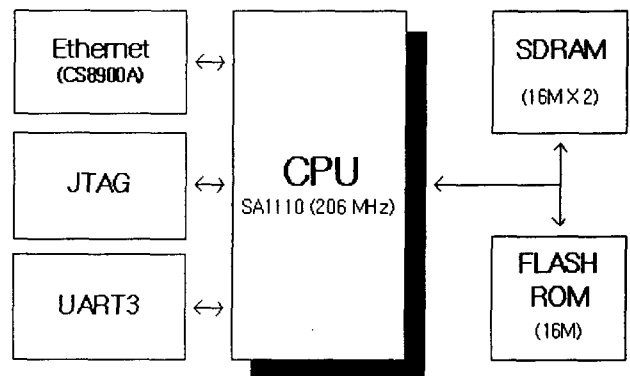


그림 2 시스템 구성도

## 2.2 Strong ARM SA1110 Processor

32비트 RISC방식의 프로세서이며 ARM V4의 코어를 사용하고 있다. 낮은 소비전력과 매우 작은 패키지 크기를 갖추어서 PDA와 같은 소형화 기기에 유리하며 3.3V~6V I/O입출력, 클럭 발생기 내장, 전원 제어 모드, MMU(메모리 관리 유닛), 명령캐쉬(16K byte), 데이터 캐쉬(8K byte) 기능, 28개의 인터럽트 처리 가능한 GPIO등의 기능을 갖추고 있다. 이처럼 SA1110는 MPU와 그 외 주변 장치들이 하나의 패키지 형태로 집적되어 있는 embedded 시스템이다.

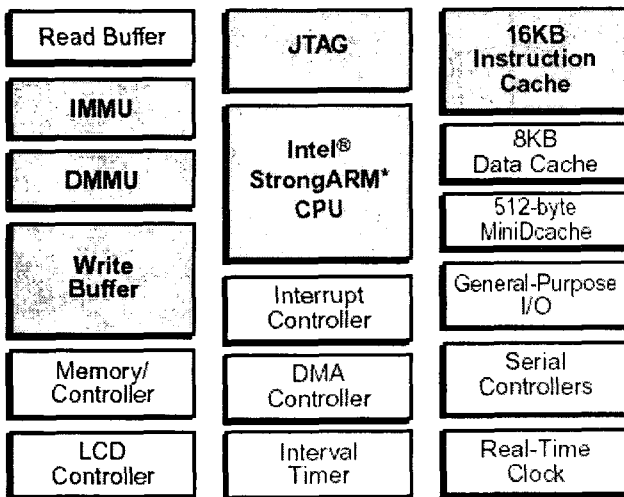


그림 3 StrongARM SA1110 Processor

## 2.3 Embedded Linux 시스템

LINUX는 Unix의 운영체제로 가장 큰 장점은 소스코드가 개방되어 있다는 것이다. 이것은 새로운 S/W의 연구나 개발을 촉진하고 운영체제 확장을 가능하게 한다. 가격 경쟁력 또한 우수하며 다양한 서비스를 제공하여 사용자가 곧 개발자가 될 수 있어 오류가 생길시 디버깅이 빠르기 때문에 그만큼 안정적이며 다양한 응용에 따른 성장가능성이 높다.

### 2.3.1 Toolchain

Toolchain은 Embedded 시스템의 특성상 큰 용량의 저장 장치를 가지지 못하기 때문에 호스트 시스템에서 개발을 하기 위해 필요한 모든 환경을 말한다. SA1110 MPU를 사용하기 위하여 ARM cross 컴파일러를 사용하게 된다. 컴파일러는 GNU C 컴파일러를 사용한다. 일반 GNU Tool과는 다르게 컴파일을 하게 되면 ARM에서 실행 가능한 바이너리가 생성되는 cross 컴파일러이다. Cross 컴파일러를 이용하여 바이너리(boot-loader, 커널...)를

생성한 후 직렬 포트 혹은 JTAG를 이용하여 타겟 보드에 다운로드 한다.

### 2.3.2 Boot-loader 포팅

CPU가 리셋 신호를 받고 나면 플래쉬 메모리에서 명령어를 읽어서 동작을 시작한다. 플래쉬 메모리에 boot-loader가 존재한다. boot-loader는 하드웨어 (MPU, SDRAM, 플래쉬, UART)등을 초기화하며 리눅스를 부팅(커널이미지를 SDRAM에 저장한 후 커널 이미지의 주소로 점프)하고 커널과 파일 시스템을 다운로드하고 플래쉬에 쓰는 기능을 수행한다.

Blob을 컴파일 하기 위해서는 우선 ToolChain이 설치되어 있어야 하며 이를 플래쉬에 직접 쓰기 위하여 JTAG를 사용한다.

Flash(16MB, 16bit)

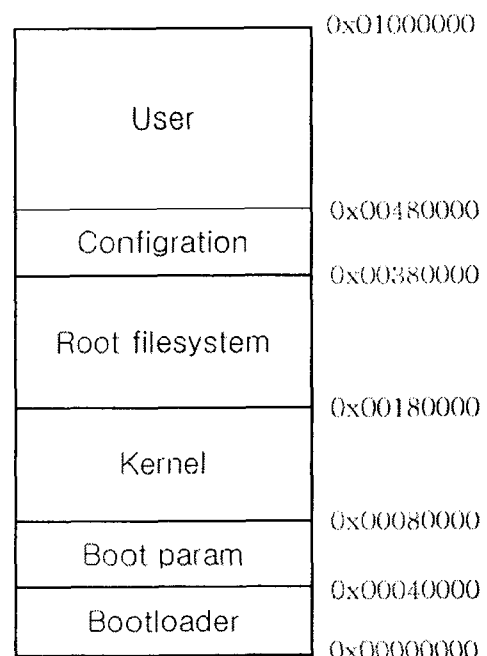


그림 4 플래쉬 메모리 맵

### 2.3.3 커널 포팅

커널은 운영체제의 핵심으로 프로세서(메모리에서 실행중인 프로그램)관리, 입출력 작업관리 등을 담당한다. 웹상의 많은 커널 중 본 연구에서는 커널2.4.5를 사용했으며 ARM, SA1110용으로 Patch 하여 사용한다. 커널 컴파일은 일반 리눅스에서의 컴파일과 크게 다르지 않으며 xconfig를 하여 필요한 Module을 포함시켜 의존도를 검사한다. 이후 zImage를 만들어 Blob을 이용하여 SDRAM에 다운로드하고 이를 Blob의 플래쉬 명령을 사용하여 플래쉬의 0x00080000 번지에 저장한다.

### 2.3.4 파일 시스템

Embedded Linux에서 램 디스크, JFFS, JFFS2, Cramfs, Ramfs등을 사용할 수 있다.

램 디스크는 메모리의 일부를 디스크로 인식시킨 것이며 가장 일반적으로 램 디스크를 Root Filesystem으로 사용한다. 램 디스크는 Ram에서 동작하기 때문에, 속도가 빠를 뿐만 아니라 Gzip압축을 사용하여 용량을 줄일 수 있다는 장점이 있다. JFFS(Journalling Flash File System)는 플래쉬에서 구현되는 파일시스템으로 보통 램 디스크와 같이 구현이 되며, 저장이 되어야하는 내용들을 JFFS를 구현하여 사용한다. 이러한 파일시스템을 이용하여 만들어진 이미지를 Blob을 이용하여 다운로드, 플래쉬에 저장하면 리눅스로 부팅 할 수 있게 된다.

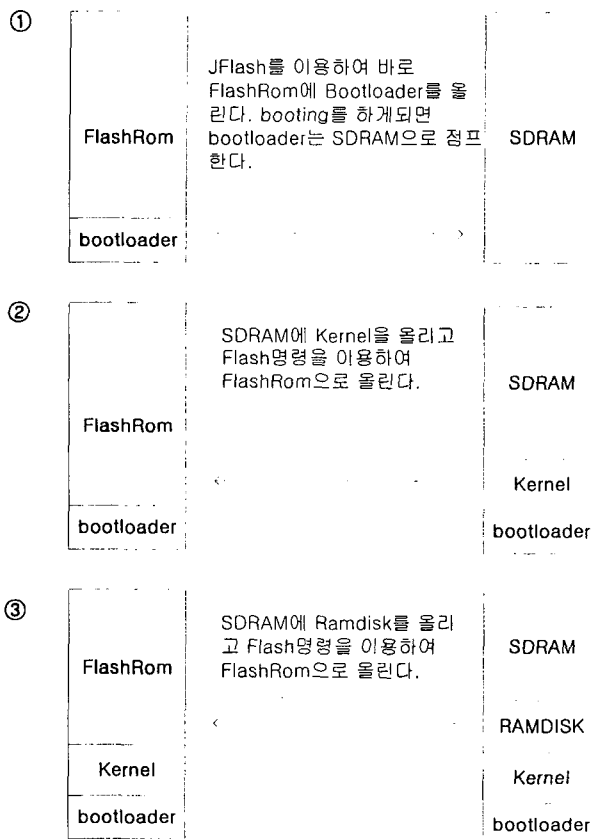


그림 5 플래쉬 포팅 과정

### 2.4 터미널 서버

일반적으로 정보기술에서 말하는 터미널 서버는, PC나 프린터 기타 주변장치 등과 같은 터미널들이 근거리통신망, 또는 광역통신망에 접속할 수 있는 포트를 제공하는 하드웨어 장치이다. 터미널들은 RS-232C 또는 RS-423 등과 같은 직렬포트를 이용하여 터미널 서버에 접속한다. 터미널 서버 내부의 접속자원은 접속되어 있는 모든 터미널들에 의

해 동적으로 공유된다. 일부 터미널 서버들은 최대 128대까지의 터미널을 지원할 수 있다. 그러한 터미널들에는 PC나 3270 터미널, 프린터, 기타 RS-232/423 인터페이스를 가지는 여러 가지 주변장치들이 포함된다. 터미널들은 호스트에 텔넷 접속을 하기 위해 TCP/IP 프로토콜을 사용한다. 터미널 서버에서는 터미널 사용자가 유닉스, IBM, DEC 등 여러 가지 다른 종류의 호스트 운영체계에 다중 접속을 할 수 있도록 허용하기도 한다. 본 연구에서는 Strong ARM Board의 직렬 포트를 RS232C로 연결하여 터미널 서버를 구현하고자 한다.

#### 2.4.1 TCP 서버

서버의 일은 통신의 종단점을 설정해 놓고 클라이언트로부터의 연결 요구를 수동적으로 기다리는 것이다. 먼저 socket() 루틴을 사용하여 TCP/IP socket을 생성한다. 그리고 소켓과 결합할 포트에 bind()로 소켓에 한 포트 번호를 부여한다. 시스템에 listen()을 이용 그 포트에 연결 요구를 허락하도록 알린다. 그 뒤에 다음 과정을 반복한다. 각 클라이언트 연결에 대해 새로운 소켓을 얻기 위해 accept()를 호출한다. send()와 recv()를 사용 그 새로운 소켓을 통해 클라이언트와 통신한다. close()를 사용하여 해당 클라이언트와의 연결을 닫는다. 소켓을 생성하고, 보내고, 받고, 닫는 것들은 클라이언트에서와 마찬가지로이다.

서버에서의 차이점은 주소를 소켓에 결합(bind)하고 그 소켓을 클라이언트들에 연결된 다른 소켓들로부터 “받는” 채널로 사용한다.

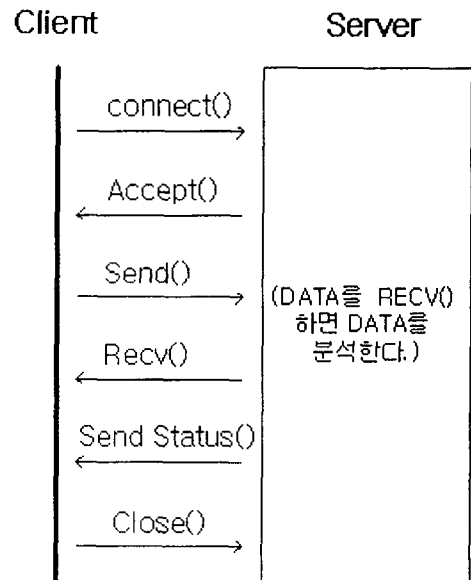


그림 6 Client 와 서버

## 2.4.2 TCP 클라이언트

클라이언트와 서버와의 큰 차이점은 접속하기 전에 상대방의 소켓에 연결(Connect)이 되어야 하며 만약 연결이 설정될 수 없으면 나중에 다시 시도해야 한다. 서버는 클라이언트로부터 연결 설정을 수동적으로 기다리는 반해, 클라이언트는 연결 설정을 시작한다. 클라이언트의 일은 접속되기를 수동적으로 기다리는 서버와 통신을 시작하는 것이다. socket()을 사용하여 TCP소켓을 생성하고 connect()를 사용하여 서버와 연결을 설정한다. 그런 뒤 send()와 recv()를 사용하여 통신한 뒤 통신이 끝난 후 close()로 연결을 닫는다.

## 3. 결 론

LINUX는 그 동안 데스크탑에서의 풍부한 개발 경험으로 가격 경쟁력, 성능의 우월성 및 개발의 용이성 등에서 검증된 운영체제이다. 이와 같은 리눅스 운영체제의 장점들이 Embedded 시스템의 개발에도 그대로 적용되고 있다. 본 논문에서도 Embedded LINUX를 포팅하여 서버를 만들고 다른 PC에서 Client로 접속하여 Strong ARM Board에 RS232C로 연결된 직렬 포트를 TCP/IP Control과 직렬 포트의 수를 늘여 터미널 서버를 구현한다.

## 참 고 문 헌

- [1] Michael Barr, "Programming Embedded 시스템: In C And C++", 1999, June.
- [2] Intel StrongARM SA-1110 Microprocessor, "Developer's Manual", 2000, June.
- [3] Donahoo/Calvert, "The Pocket Guide to TCP/IP Sockets", 2001, October
- [4] Tomohiro Odaka, "TCP/IP 에널라이저 작성과 패킷 해석(Linux/FreeBSD)", 2002, January
- [5] Rubini, "LINUX CEVICE DRIVERS", 2000, Jun.
- [6] Daniel P. Bovet/Marco Cesati, "Understand LINUX 커널" 1999, Jun
- [7] Warren Gay, "C 사용자를 위한 리눅스 프로그래밍",