

# ITRON 개발환경

## 또 다른 오픈 소스 임베디드 ITRON ②

현재 일본의 경우 약 9만5,000명 정도 임베디드 소프트웨어 인력이 부족한 상태로, 점점 일본 젊은 층 인구가 줄고 있는 상황에서 임베디드 분야의 인력 부족 현상은 앞으로 더욱 심화될 것으로 예상된다. 본 연재에서 다룰 ITRON은 일본에서 가장 많이 쓰는 Hard Real Time 임베디드 OS로서, 일본 임베디드 산업체에 취업을 고려 중인 독자들에게 조금이나마 도움이 되었으면 한다. 또한 공개 소스 형태인 ITRON을 소개함으로써 국내 기업들도 무료인 Hard Real Time 임베디드 OS를 제품 개발에 적극 활용할 수 있기를 바라는 마음이다.

글 : 채승엽 / Navithes(나비더스)  
sychae@navithes.com

지난 1회 연재에서는 ITRON에 대한 소개와 함께 관련된 전반적인 배경에 대해 알아보고, 2회부터는 본격적으로 ITRON의 개발환경에 대해서 설명을 하겠다.

앞으로의 설명은 TOPPERS(<http://www.toppers.jp>)에서 공개된 TOPPERS/JSP(Release 1.4.3)인 ITRON 4.0을 기준으로 진행될 것이다. 관련 자료는 TOPPERS 홈페이지에서 TOPPERS/JSP(<http://www.toppers.jp/jsp-download.html>)를 누구나 다운로드 받을 수 있다.

### 약어 설명

- TOPPERS : “Toyohashi OPen Platform for Embedded Real-time Systems”
- JSP : “Just Standard Profile”
- FI4 : “Fullset ITRON4”
- FDMP : “Function Distributed Multi Process”
- OSEK : “Offene Systeme und deren Schnittstellen für die Elektronik im

Kraftfahrzeug”(차재 엘렉트로닉스를 위한 오픈 시스템과 통신 인터페이스)

- MPU : “Micro Process Unit”, 일본에서는 임베디드용 목적의 SoC된 CPU를 MPU라고 함.

### 필요한 문서들

- ITRON 4.02 사양서

일본어 : <http://www.ertl.jp/ITRON/SPEC/mitron4-j.html>

영어 : <http://www.ertl.jp/ITRON/SPEC/mitron4-e.html>

- 최신 ITRON 4.03 사양서

**\* 연재순서**

**또 다른 오픈 소스 임베디드 ITRON**

- 1화: ITRON 소개 및 제품 사용현황
- 2화: ITRON 개발환경
- 3화: XScale보드에 ITRON포팅하기 ①
- 4화: XScale보드에 ITRON포팅하기 ②
- 5화: x86에 ITRON 시뮬레이션으로 application 작성하기
- 6화: ITRON에서 device driver 개발하기
- 7화: ITRON에서 application 개발하기

/필/자/스/개/

채승엽(sychae@navithes.com)



2005년 일본 T-Engine포럼 내 UNL(Ubiquitous Networking Laboratory)연구소에서 ITron 및 T-Engine 전문 강사 과정을 연 수 받았으며 현재 일본의 임베디드 기술 관련하여 연구 하고 있다. 그리고 T-Engine 포럼의 한국어 웹사이트인 KTEC(http://www.t-engine.or.kr)에 기술 칼럼을 연재하고 있다.

일본어 : <http://www.assoc.tron.org/spec/itron/itron403/itron403.htm>

- ITRON의 API TCP/IP 1.0 사양서

일본어 : <http://www.ertl.jp/ITRON/SPEC/tcpip-j.html>

영어 : <http://www.ertl.jp/ITRON/SPEC/tcpip-e.html>

- 최신 ITRON의 API TCP/IP 2.0 사양서

일본어 : <http://www.assoc.tron.org/spec/itron/tcpip200.pdf>

**TOPPERS 프로젝트에서 공개하는 성과물들**

TOPPERS 프로젝트에서의 지금까지 개발 성과물과 관련, 표 1, 2와 같이 공개 소프트웨어를 무상으로 배포하고 있다.

**TOPPERS 기타 공개 소프트웨어**

**Litron 프로젝트 (Linux + iTRON)**

이러한 형태를 듀얼 OS 또는 Hybrid OS라고 말하며 현재 Litron ver1.3.0(Linux2.6.10 + jsp1.4)이 공개되고 있다. 관련 소스는 <http://www.goodssoft.jp/litron/> 에서 구할 수 있다.

리얼 타임 커널과 멀티미디어 기능 등의 리눅스를 하나의 MPU 위에서 구동하여, 고속 MPU에서 하드 리얼 타임 제어가 필요한 제품과 리눅스에서 제공하는 다양한 UI Application을 사용하며, 하드 리얼 타임 장치의 데이터를 데이터베이스로 구축할 수 있다.

현재 도요타에서는 이와 비슷한 원리의 운영체제인 'H2O' 라는 자회사 운영체제를 2007년에 1.0Ver 개발 완료하였으며, ECU(자동차 전장 부품)과 카 내비게이션 등의 멀티미디어 기능을 하나의 고속 MPU(1GHz이상)에서 제어하는 것을 실제로 자동차에 적용하고 있다.

ITRON과 Linux가 공개 소프트웨어이므로 ITRON에서는 하

TOPPERS/JSP	μITRON4.0 사양의 standard profile 규정에 따른 리얼타임 커널로 TOPPERS 프로젝트의 최초 개발 성과물이다. 최신의 버전에서는, 약 15 종류의 프로세서를 지원하고 있으며, 많은 기업들에 의해, 각종 제품이나 prototype 개발에 활용되고 있다. 또한 MS Windows와 Linux에서 동작하는 ITRON 시뮬레이션 개발환경을 제공하고 있다.
TOPPERS/F4	JSP 커널을 기초로, μITRON4.0 사양에 기능을 더 확장한 리얼타임 커널이다.
TOPPERS/OSEK	자동차 제어 시스템 분야에서의 국제 표준인 OSEK/VDX OS사양을 기준으로 하고 있는 리얼타임 커널. (ITRON과 관계없음.) 실제 자동차에 적용하여 각종 전시회에서 시연하고 있음.
TOPPERS/FDMP	JSP 커널을 기초로, μITRON4.0 사양에 분산형 멀티프로세서 전용으로 여러 개의 MPU 위에서 동작하는 분산형 리얼타임 커널.

표 1. TOPPERS에 공개된 임베디드 오픈 소스 운영체제 종류

TINET	ITRON TCP/IP API 사양에 준거한 TCP/IP 프로토콜 스택으로 JSP커널과 F4커널에서 동작하며, IPv6의 기능도 포함 되어 있다.
FatFs for TOPPERS	FAT 파일 시스템. FAT 12/16/32에 대응하고 있으며, μITRON4.0 에서 MMC, CF카드, 하드 디스크 등을 작동할 수 있다.
RLL (Remote Link Loader)	리눅스 서버 등에서 저장되어 있는 임베디드용 소프트웨어 모듈을, 임베디드 제품에 동적으로 추가, 삭제, 갱신하기 위한 ITRON 미들웨어다. 임베디드용 소프트웨어 모듈의 링크를 서버측에서 작동하여 필요한 소프트웨어 모듈을 모아 두어 임베디드 제품에 빠르게 다운로드하며 전송시 오버헤드가 작다. 임베디드 제품에는 RLL Agent 미들웨어, 리눅스 서버에는 RLL 서버가 동작하여 서비스가 이루어진다.
DLM (Dynamic Loading Manager)	임베디드 제품의 파일 시스템에 저장 되어 있는 파일을 읽어 들여서 RAM에서 다 로드하여 실행을 한다. 새로운 어플리케이션과 라이브러리는 설치 할 수 있다. 휴대전화나 PDA에서 어플리케이션을 설치 및 업그레이드 할 때 사용되는 기술이다. 서버가 없어도 임베디드 제품에서 독립적으로 작동이 할 수 있다.
TOPPERS C++ API 템플릿 라이브러리	μITRON 사양 준거하여 커널에서 사용할 수 있는 C++용 템플릿 라이브러리며, 기본적으로 μITRON에서는 C++을 지원하지 않는다. 이것을 추가함으로 C++ 사용할 수 있다.
모델 베이스 개발 지원	Toyo회사에서 만든 BridgePoint 프로그램으로 생성한 코드를, JSP 커널에서 동작시키기 위한 런타임 및 톨, 쉽게 말하면 임베디드 제품에서 μITRON 동작을 소스 레벨로 실시간 분석을 할 수 있다.

표 2. TOPPERS에서 공개된 TOPPERS의 ITRON에서 작동되는 오픈 소스 미들웨어

드 리얼 타임 문제를 해결하고 리눅스에서는 UI Application과 데이터베이스 문제를 해결한다.

TOPPERS/JSP에서 지원하는 MPU 종류  
(jsp/config 디렉터리)

디렉터리 명	개발 환경
MPU	시스템 (회사명)
m68k	GNU 개발환경
M68040 (MC68LC040)	DVE-68K/40 (電産)
sh1	GNU 개발환경
SH1 (SH7032)	KZ-SH1-01 (京 都 マイコンピョタ)
	※ RISC평가 키트 SH-1 (CQ출판사도 작동)
SH1 (SH7034)	μTRON 탑재 SH1CPU보드 (株)中央製作所
sh2	GNU 개발환경
SH2 (SH7145)	AP_SH2F_6A (알파 프로젝트 회사)
SH2	(SH7615)HSB7615IT (北斗電子)
sh3	GNU 개발환경
SH3 (SH7709A)	MS7709ASE01 (Hitachi System LSI)
SH3 (SH7729R)	MS7729RSE01 (Hitachi System LSI)
SH3SH7727)	MS7727CP01 (Hitachi System LSI)
SH4 (SH7750)	MS7750SE01 (Hitachi System LSI)
sh3-ghs	GHS 개발환경
SH3 (SH7709A)	MS7709ASE01 (Hitachi System LSI)
SH3SH7727)	MS7727CP01 (Hitachi System LSI)
h8	GNU 개발환경
H8 (H8/3052F)	AKI-H8/3052F (秋月電子通商)
H8 (H8/3069F)	AKI-H8/3069F (秋月電子通?)
h8-renesas	Renesas의 개발환경
H8 (HSB8F3048BF25)	H8/3048F-ONES ㄸ 키 ッ (株)北斗電子
armv4	GNU 개발환경
ARM9 (ARM922T)	KZ-ARM9EXPCI-01 (京 都 マイコンピョタ)
ARM9 (ARM926EJ-S)	AZ9360MB (YDK)
armv4-ghs	GHS 개발환경
ARM9 (ARM920T)	Integrator/AP+CM920T (ARM)
ARM9E (ARM966E-S)	Integrator/AP+CM966E-S (ARM)
m32r	GNU 개발환경
M32R (M32102S6FP)	M3A-2131G50 (미츠비씨전기)
M32R (M32102S6FP)	M3A-ZA36 (미츠비씨전기)
m32c-renesas	Renesas의 개발환경
M32C	OAKS32 (ㄸ ㄸ 電 子)
microblaze	GNU 개발환경
MicroBlaze	MIREF (YDK)
MicroBlaze	MIRE_MULT3000 (YDK)
MicroBlaze	MultiMedia Board (Xilinx)
MicroBlaze	Suzaku (Atmark)
tms320c54x	TI의 개발환경
TMS320C54x (TSM320C5402)	TMS320VC5402 DSK (TI)
xstormy16	GNU 개발환경
xstormy16	산요 개발 톨 (산요반도체)
m16c-renesas	Renesas의 개발환경
M16C (M30262F8FG-CPU)	OAKS16 (ㄸ ㄸ 電 子)

	M16C (M30262F8FG-CPU)	OAKS16-MINI (ㄸ ㄸ 電 子)
s1c33		GNU 개발환경
	SC33	DMT33209 (EPSON)
	SC33	LUXUN2 (EPSON)
s1c33-gnu33		GNU33 개발환경
	SC33	DMT33209 (EPSON)
	SC33	LUXUN2 (EPSON)
nios2		GNU 개발환경
	Nios2	NiosII Development Board (Atela)
v850 GNU		개발환경
	V850	TK-850/KJ1+ (Application Corp.)
	V850	TK-850/SG2 (Application Corp.)
tlcs900		도시바의 개발환경 (TOSHIBA IDE)
	TMP91CY22-CPU	Zup-F16화장 보드 (ㄸ ㄸ 電 業 (株))

대표적으로 도시바, EPSON, Hitachi, 산요, 마츠비씨 등에서 TOPPERS/JSP를 제품 개발에 많이 사용하고 있다는 것을 알 수 있다.

### 개발 환경

JSP 커널은 GCC 등의 GNU 개발 환경을 표준으로 하고 있지만, 다른 종류의 개발 환경도 이용할 수 있도록 고려하고 있다. 이용할 수 있는 개발 환경에 대해서는, 타깃 보드 마다 또는 개발 환경 마다 매뉴얼(jsp/doc 디렉터리)로 설명한다.

그리고 JSP 커널에서는 많은 부분을 C언어로 작성되어 있으며, 타깃 보드 의존부와 CPU 의존부를 명확하게 분리하여 다른 Target 보드에 포팅이 용이하도록 소스 디렉터리가 구성되어 있다.

JSP 커널의 특징에서 대해서 상세하게 설명을 하면 좋겠으나 잡지에 연재되는 페이지 수의 제한이 있으므로 큰 특징만을 설명 하겠다.

JSP 커널은 μTRON4.0 사양의 스탠다드 프로파일에 포함되는 기능을 모두 지원하고 있다. 스탠다드 프로파일에서는, 인터럽트 핸들러와 인터럽트 서비스 루틴을 둘 다 지원하면 좋겠지만, JSP 커널은 현재 인터럽트 핸들러만을 지원하고 있다.

그리고 제일 큰 특징은 ITRON 사양의 변수 표현 방식이다.

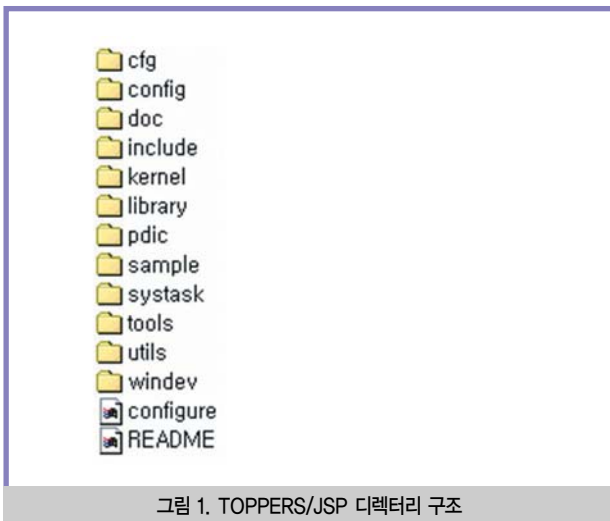


그림 1. TOPPERS/JSP 디렉터리 구조

ITRON 사양을 따르는 운영체제들은 아래와 같은 변수 표현 방식을 사용함으로 주의하자. 물론 기존의 표준 C언어의 변수 표기 방법을 사용해도 관계는 없다. 그러나 ITRON 사양의 변수 표현 방식을 권장한다.

jsp/include/itron.h에 정의되어 있는 ITRON 사양 공통 데이터형 표현 방식

```
typedef signed_int8_      B;    /* 부호 첨부 8 비트 정수 */
typedef unsigned_int8_   UB;   /* 부호 없음 8 비트 정수 */
typedef_int8_            VB;   /* 형이 정해지지 않는 8 비트의 값 */

typedef signed_int16_    H;    /* 부호 첨부 16 비트 정수 */
typedef unsigned_int16_ UH;   /* 부호 없음 16 비트 정수 */
typedef_int16_          VH;   /* 형이 정해지지 않는 16 비트의 값 */
/

typedef signed_int32_    W;    /* 부호 첨부 32 비트 정수 */
typedef unsigned_int32_ UW;   /* 부호 없음 32 비트 정수 */
typedef_int32_          VW;   /* 형이 정해지지 않는 32 비트의 값 */
typedef signed_int64_    D;    /* 부호 첨부 64 비트 정수 */
```

```
typedef unsigned_int64_  UD;   /* 부호 없음 64 비트 정수 */
typedef_int64_          VD;   /* 형이 정해지지 않는 64 비트의 값 */
/

typedef void            *VP;   /* 형이 정해지지 않는 포인터 */
typedef void            (*FP); /* 프로그램의 포인터 */

typedef signed_int      INT;   /* 부호 첨부 정수 */
typedef unsigned_int    UINT;  /* 부호 없음 정수 */

typedef_bool_          BOOL;  /* 참과 거짓 */

typedef_INT            FN;    /* 기능 코드 */
typedef_INT            ER;    /* 에러 코드 */
typedef_INT            ID;    /* 오브젝트의 ID번호 */
typedef_UINT           ATR;    /* 오브젝트의 속성 */
typedef_UINT           STAT;  /* 오브젝트 상태 */
typedef_UINT           MODE;  /* 서비스 콜의 동작 모드 */
typedef_INT            PRI;   /* 우선도 */
typedef_size_t         SIZE;  /* 메모리 영역의 사이즈 */

typedef_INT            TMO;   /* 타임아웃 지정 */
typedef_UINT           RELTIM; /* 상대 시간 */
typedef_UW             SYSTM; /* 시스템 시간 */

typedef_vp_int_        VP_INT; /* VP 또는 INT */
typedef_VP             VP_INT; /* VP 또는 INT */

typedef_INT            ER_BOOL; /* ER 또는 BOOL */
typedef_INT            ER_ID;   /* ER 또는 ID */
typedef_INT            ER_UINT; /* ER 또는 UINT */
```

char a = 'T';	UB a = 'T';
int a, b, c;	W a, b, c;
c = a+b;	c = a+b;
typedef struct {	typedef struct {
char *name;	UB *name;
int *b;	W *b;
} number;	} number;

표 3. 기존 C언어(왼쪽)의 사용과 ITRON 변수 표현 방식(오른쪽) 비교

user.txt	TOPPERS/JSP 커널 사용자 매뉴얼
gnu_install.txt	GNU 개발 환경 구축 매뉴얼
sh3.txt	타깃 의존부 매뉴얼
armv4.txt	ARMV4 타깃 의존부 매뉴얼
linux.txt	Linux 시뮬레이션 환경 의존부 매뉴얼
windows.txt	시뮬레이션 환경 의존부 매뉴얼
config.txt	JSP 커널 타깃 의존부 포팅 가이드
configurator.tx	JSP 커널 configurator 사양
design.txt	JSP 커널 설계 노트

표 4. jsp/doc/ 디렉터리 중요 파일 설명

JSP 커널에 관한 개발환경에는 아래와 같은 툴이 필요하다.

- BINUTILS : 2.10, 1
  - GCC-CORE : 2.95, 3 (GCC 컴파일러)
  - NEWLIB : 1.9, 0 (표준 C 라이브러리)
  - GDB : 4.18 (GDB 디버깅)
- 또는
- BINUTILS : 2.14
  - GCC-CORE : 3.3 (GCC 컴파일러)
  - NEWLIB : 1.11, 0 (표준 C 라이브러리)
  - GDB : 5.3 (GDB 디버깅)

GNU 개발 환경의 인스톨 방법은 임베디드 리눅스의 크로스 컴파일러를 만든 것과 동일하다. 그러므로 반도체 업체나 보드 판매 업체에서 제공하는 크로스 컴파일러를 사용하면 되겠다. 실제로 임베디드 제품을 개발할 때는 크로스 컴파일러의 안정화가 완료된 것을 구해서 사용하며, 크로스 컴파일러를 직접 컴파일해서 만들지는 않는다.

그러므로 크로스 컴파일러 만드는 방법에 대해서는 생략하고 실무적인 내용으로 설명을 하겠다.

주의사항으로 일반적으로 크로스 컴파일러를 만들 때 사용할 커널 버전과 상관관계가 있으므로 꼭 JSP커널 소스의 'jsp/doc/'

에서 해당되는 개발 CPU에 대한 개발환경을 반드시 읽어 보기 바란다. 읽어 보지 않으면 make를 사용하여 커널 이미지를 만드는 과정에서 에러가 발생한다.

실제로 TOPPERS/JSP 커널을 개발하기 위한 파일을 설명하면, 반도체 업체나 보드 판매 업체에서 제공하는 안정화된 크로스 컴파일러와 NEWLIB만 있으면 되겠다. 아니면 안정화된 크로스 컴파일러에 NEWLIB도 설치가 완료된 크로스 컴파일러를 구하면 되겠다.

먼저 TOPPERS/JSP에서 예제로 제시하고 있는 SH(Super Hitachi) MPU인 SH7727의 T-Engine 보드에 TOPPERS/JSP 커널을 만드는 방법을 서술하겠다. 본 개발 방법은 Cygwin에서 개발하는 방법을 기준으로 설명을 하겠다. TOPPERS/JSP가 용량이 아주 작으므로 MS Windows에서 Cygwin에서 개발하면 되겠다. 그리고 Cygwin 설치 방법은 쉽기 때문에 설명을 생략하겠다.

### 필요한 파일들

- Cygwin 셋업 프로그램 (setup.exe)  
<http://sources.redhat.com/cygwin/>의 홈페이지에서 다운로드하여 전체 설치를 하면 된다.
- GNU 개발 환경 바이너리 패키지 (gnu\_cygwin\_sh.tar.gz)  
<http://www.toppers.jp/jsp-kernel.html>의 홈페이지에서 '개발 환경 바이너리 패키지(SH, Cygwin)' 을 클릭해서 다운로드 한다.
- NEWLIB Ver.1.9.0 (newlib-1, 9.0.tar.gz)  
<http://sources.redhat.com/newlib/>의 홈페이지에 접속해서 다운로드 한다.
- TOPPERS/JSP 커널 다운 받기 (jsp-1.4.3.tar.gz)  
<http://www.toppers.jp/jsp-download.html>의 홈페이지에 접속해서 다운로드 한다.

## GNU 개발 환경의 구축 및 커널 이미지 만들기

### 1. gnu\_cygwin\_sh.tar.gz 설치

'/usr/local'의 경로에 'gnu\_cygwin\_sh.tar.gz'을 압축을 풀면 된다. 그 결과 '/usr/local/sh'라고 하는 디렉터리가 생성된다.

```
$ cd /usr/local
$ tar zxvf gnu_cygwin_sh.tar.gz
```

### 2. 크로스 컴파일의 경로를 패스를 해준다.

다음과 같이 홈 디렉터리로 이동하여 '.bash\_profile'이 있는지 확인한다.

```
$ cd $HOME
$ ls -f
```

홈 디렉터리에 있는 '.bash\_profile'을 vi 등으로 열어서 맨 아래 줄에 다음과 같이 PATH를 작성한다.

```
export PATH=/usr/local/sh/bin:${PATH}
```

그리고 source라는 명령어로 패스 경로를 바로 적용시켜 준다. 그렇게 하면 크로스 컴파일러를 어느 디렉터리에서도 사용할 수 있다.

```
source .bash_profile
```

### 3. newlib-1.9.0.tar.gz을 설치하기

'/usr/local'로 이동하여 'source'라고 하는 임의의 디렉터리를 새롭게 만든다.

그리고 newlib-1.9.0.tar.gz의 압축을 풀면 '/usr/local/source/newlib-1.9.0'라고 하는 디렉터리가 만들어 진다. 그리고 바로 'newlib'라고 하는 임의 디렉터리를 만들어 SH용 newlib를 만든다.

```
$ cd /usr/local
$ mkdir source
$ tar zxvf newlib-1.9.0.tar.gz
$ mkdir newlib
$ cd newlib
$ ../newlib-1.9.0/configure --target=sh-hitachi-elf --
prefix=/usr/local/sh
$ make
$ make install
```

\* newlib는 레드햇에서 만든 glibc를 축소시킨 표준 C언어 라이브러리이다.

\* --target은 해당 MPU를 지정하고, --prefix는 만들어진 라이브러리를 'make install'의 명령어를 이용해 복사할 경로 위치이다.

여기서 정확한 복사 위치는 /usr/local/sh/sh-hitachi-elf의 디렉터리 위치에 newlib에서 생성한 파일을 include와 lib에 복사해 준다.

### 4. TOPPERS/JSP 커널 이미지 만들기

홈 디렉터리로 'jsp.tar.gz'를 압축을 풀면 'jsp' 디렉터리가 만들어진다.

```
$ cd $HOME
$ tar zxvf jsp-1.4.3.tar.gz
```

이 상황에서 TOPPERS/JSP를 컴파일 하기 위해 배치 틀 환경을 구축한다.

```
$ cd jsp/cfg
$ make depend
$ make
```

여기서 홈 디렉터리에 있는 'jsp' 폴더로 이동하여 SH7727의 T-Engine 보드용 커널 이미지를 만들어 본다. 커널 이미지를 생성할 디렉터리 이름을 임의로 'test'로 정했다.

```
$ cd $HOME
$ cd jsp
$ mkdir test
$ cd test
$ perl ../configure -C sh3 -S solution_engine
```

\* -C sh3 : -C는 Target MPU의 이름이며, 뒤에 sh3를 입력하여 'jsp/config/sh3'의 경로를 말한다. 그러므로 'jsp/config/' 아래의 폴더명과 동일한 이름이다.

\* -S solution\_engine : -S Target 보드의 이름이며, solution\_engine를 입력하여 'jsp/config/sh3/solution\_engine'의 경로를 말한다. 나중에 XScale 보드에 TOPPERS/JSP 커널을 포팅할 때 이해하고 있어야 할 개발 환경이다.

위의 작업에 의해서 'test' 디렉터리에는 샘플 프로그램

(jsp/sample 디렉터리)의 'sample.c', 'sample.h', 'sample.cfg'와 'Makefile'가 만들어지고, 'jsp/cfg'에는 컴파일러(cf.exe) 파일이 생성된다.

여기서 아래와 같이 make 명령어를 사용하면 커널 이미지가 만들어진다.

```
$ make depend
$ make
```

커널 이미지 파일은 'jsp.srec'이며 모토로라 S-Format의 이미지이다.

참고로 일반적으로 우리나라에서는 binary 파일 커널 이미지를 많이 사용하고 있지만 일본에서는 모토로라 S-Format을 많이 쓴다.

S-Format의 장점은 시리얼로 보드에 커널 이미지를 전송할 때 파일 전송 에러를 검증을 해주는 것이 특징이다. 아래는 리눅스 부트로더의 'u-boot'의 S-Format과 binary 형식을 만드는 방법이다. 아래의 빨간색만 수정하면 커널 이미지 형태를 쉽게 바꿀 수 있다.

```
u-boot.srec: u-boot
    $(OBJCOPY) $(OBJCFLAGS) -O srec $< $@
u-boot.bin: u-boot
    $(OBJCOPY) $(OBJCFLAGS) -O binary $< $@
```

## Eclipse를 이용한 TOPPERS/JSP 개발하기



Toppers 전용 eclipse 개발 환경 공개 틀인 PizzaFactory를 설명하

했다. PizzaFactory는 일본의 모나미소프트웨어 회사에서 만들었으며, 배포판은 무료이며 상업용 버전도 있다.

PizzaFactory3.1.1 Express(2007년)는 <http://www.pizzafactory.jp/>의 홈페이지에서 다운로드 할 수 있다.

メインメニュー-(메인 메뉴) → ダウンロード(다운로드) → 無償ソフトウェア(무상소프트웨어) → PizzaFactory3.1.1 Express (2007년)

PizzaFactory는 기존의 eclipse 개발환경을 TOPPERS 전용으로 최적화하여 만들 개발 TOOL이다. 그러므로 TOPPERS에 공개한 모든 소프트웨어의 개발이 가능하다.

PizzaFactory의 특징으로는 용량이 75MB로 아주 작고 그림 3, 4와 같이 인터넷에 접속하여 추가 기능을 다운 받을 수 있다는 점이다. 그리고 Cygwin 개발 환경보다 컴파일 속도가 약 10배 정도 빠르다.

그리고 무엇보다 큰 장점은 그림 5와 같이 모든 MPU의 TOPPERS 소프트웨어의 Cygwin용 크로스 컴파일 무료로 다운 받을 수 있다는 것이다.

그림 6처럼 PizzaFactory3에서 다운받은 크로스 컴파일러의 위치를 찾아서 Cygwin에 복사해서 TOPPERS전용 크로스 컴파일러를 사용하면 된다.

### TOPPERS/JSP의 버전 이력

- 2000년 11월 15일 Release 1.0 최초의 릴리스
- 2000년 11월 24일 Release 1.0 (PL=1) 문제점의 수정
- 2001년 2월 24일 Release 1.1 V850의 추가 등
- 2001년 5월 9일 Release 1.1 (PL=1) SH1의 추가 등
- 2001년 11월 15일 Release 1.2 SH4, H8, ARM7TDMI의 추가 등
- 2002년 4월 15일 Release 1.3 M32R, MicroBlaze, TMS320C54x, i386, H8S의 추가 등

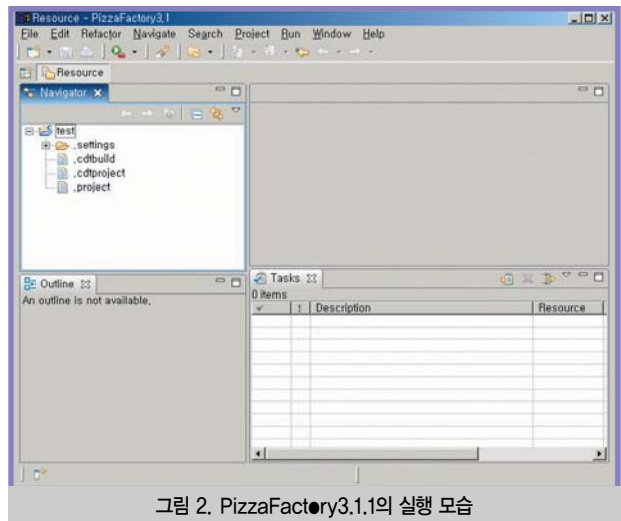


그림 2. PizzaFactory3.1.1의 실행 모습

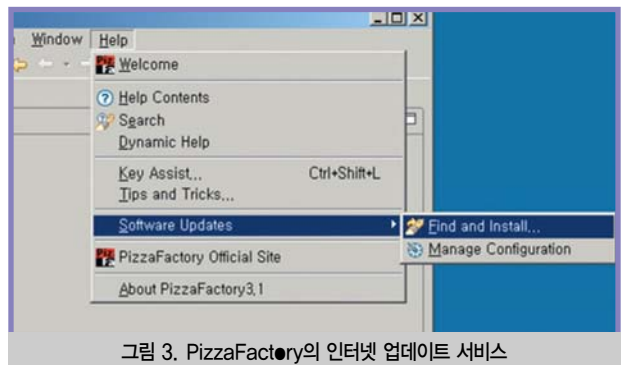


그림 3. PizzaFactory의 인터넷 업데이트 서비스

- 2003년 12월 25일 Release 1.4 다수의 개량
- 2004년 10월 14일 Release 1.4 (PL=1) SH2, M16C, SC33, PowerPC32, Nios2의 추가 등

TOPPERS의 개발된 성과물의 공개는 2년 전에 개발이 완료한 안정화 버전을 공개하고 있다. 만약 최신 자료와 현재 개발 중인 소스를 다운받으려면 회원으로 가입을 해야 한다.

회원 가입과 관련해 세부적인 내용은 표 5에 설명되어 있다. 일본의 경우 대부분의 공개 소프트웨어 관련 커뮤니티들은

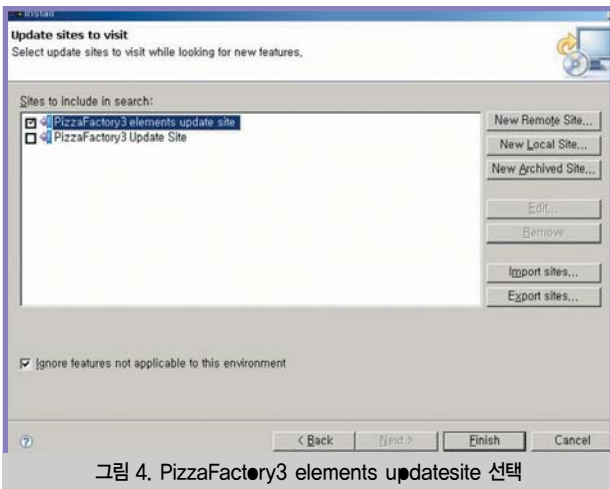


그림 4. PizzaFactory3 elements updatesite 선택

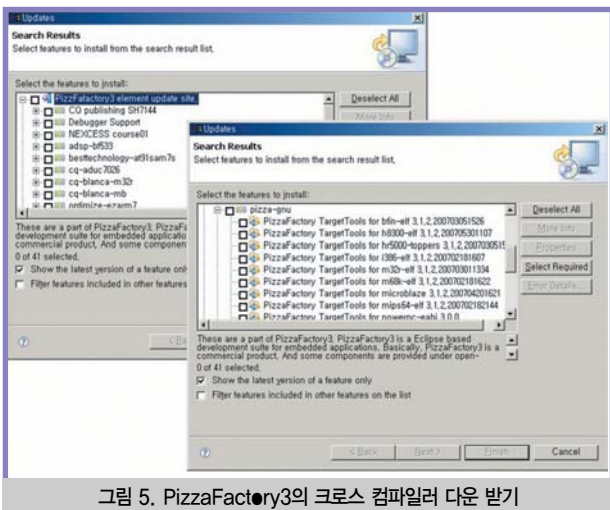


그림 5. PizzaFactory3의 크로스 컴파일러 다운 받기

NPO 법인이며 입회비와 연회비를 받고 있다. 그리고 대학교와 기업들의 참여가 활발하여 각종 임베디드 전시회에서 대기업 부스 규모로 출전하고 있다.

지금까지 TOPPERS/JSP의 개발환경에 대해서 알아보았다. 현재 일본에는 TOPPERS/JSP( $\mu$ TRON4.0) 외에  $\mu$ TRON사양

회원	자격	입회비	연회비
정회원	단체 또는 개인	단체 : 10만엔 개인 : 2만엔	단체 : 10만엔 개인 : 2만엔
준회원	개인 프로젝트에 공헌이	5000엔 없음	5000엔 없음
특별회원	있다고 인정되는 교육기관, 공적 기관, 비영리 단체,		

표 5. NPO법인 TOPPERS 회원 자격

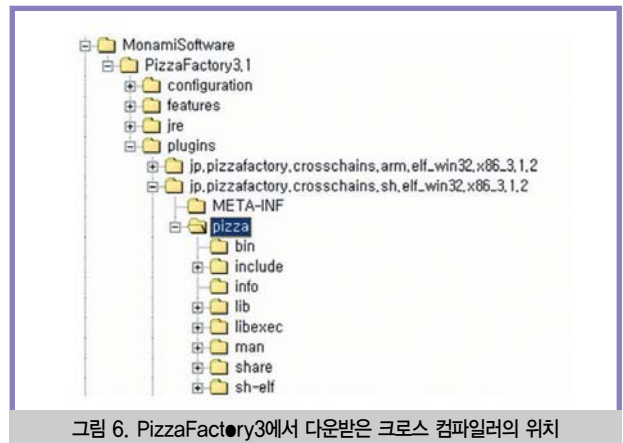



그림 6. PizzaFactory3에서 다운받은 크로스 컴파일러의 위치

기반의 다른 공개 리얼타임 운영체제도 많다. 그러나 그것은 개인이나 특정회사에서 만든 것이므로, 최근 임베디드 제품에 많이 탑재되고 있는 TOPPERS/JSP를 중심으로 설명을 전개해 나갈 생각이다.

다음 연재에는 우리나라의 임베디드 리눅스 회사인 Falinux (<http://www.falinux.com>)의 XScale 보드인 EX-5에 TOPPERS/JSP를 포팅하는 방법을 2회분에 걸쳐 연재를 하겠다. 현재 TOPPERS/JSP는 XScale이 포팅이 되어 있지 않다. 그러므로 독자들에게 가치가 있는 내용이라고 생각한다. 

관련 사이트

<http://www.tron.org>

<http://www.toppers.jp>

<http://www.t-engine.or.kr>