

특 집

임베디드 소프트웨어 개발 도구

임 채 덕*

목 차

1. 서 론
2. 임베디드 소프트웨어 개발 환경 기술 분류
3. 임베디드 소프트웨어 개발 도구 기술 동향
4. 임베디드 소프트웨어 개발 도구 시장 동향
5. 결 론

1. 서 론

임베디드 시스템(Embedded System)이란 어떤 시스템에 내장되어 특정한 목적만을 수행하는 컴퓨터이다. 쉽게 주위에서 볼 수 있는 임베디드 시스템으로는 DVD 플레이어, 셋탑박스, MP3 플레이어, 캠코더, 디지털 카메라, 냉장고, 세탁기와 같은 가전기기가 있다. 뿐만 아니라, 승강기, 항공기, 의료기기, 자동차, 공장제어, 통신 장비(교환기, 라우터 등)도 임베디드 시스템이다.

임베디드 소프트웨어란 임베디드 시스템에 탑재되는 소프트웨어를 말한다. 임베디드 시스템의 응용 분야가 다양해지면서, 예전에는 단순했던 임베디드 소프트웨어가 점점 더 복잡하고 다양한 기능을 처리하게 되었다. Time-to-Market이 생명인 임베디드 시스템 제조업자들은 임베디드 소프트웨어를 빨리 개발하여 프로젝트 기간을 단축하는데 반드시 필요한 임베디드 소프트웨어 개발 환경을 강력히 원하고 있다.

임베디드 시스템은 대개 한정된 자원을 갖고 있기 때문에 임베디드 소프트웨어 개발 환경은 대개

호스트와 타겟으로 구성된다. 타겟에는 최소한의 개발 도구만 있고, 대부분의 도구는 자원이 풍부한 호스트에 존재한다. 임베디드 소프트웨어 개발 환경은 자원이 한정된 타겟에서 실행될 프로그램을 편리하고 빨리 개발할 수 있게 해야 한다. 임베디드 소프트웨어 개발 환경은 크게 타겟 이미지 생성 도구, 소프트웨어 개발 도구, 설계 자동화 도구, 테스트 자동화 도구의 네 분야로 구분된다. 본 고에서는 네 가지 분야 중에서 소프트웨어 개발 도구의 주요 기술 및 시장 동향을 살펴본다.

2. 임베디드 소프트웨어 개발 환경 기술 분류

임베디드 소프트웨어 개발 환경은 크게 다음 네 가지 도구 분야로 나눌 수 있다.

- 타겟 이미지 생성 도구
- 소프트웨어 개발 도구
- 설계 자동화 도구
- 테스트 자동화 도구

타겟 이미지 생성 도구란 임베디드 시스템이 동작하는 데 필요한 커널, 라이브러리, 유틸리티, 응용 프로그램들 중에서 개발자가 필요한 부분만 선

* 한국전자통신연구원 임베디드 S/W 개발도구 연구팀장

택하면 최적화된 타겟 이미지를 생성해주는 도구다. 이 도구의 예로는 MS의 플랫폼 빌더와 WinXPe 스튜디오, QNX의 시스템 빌더, Lineo의 타겟 워저드, 그리고 한국전자통신연구원에서 개발한 타겟빌더가 있다.

타겟 이미지 생성 도구는 각 컴포넌트들 간의 의존성을 관리하므로, 개발자가 어떤 컴포넌트를 선택하면 그 컴포넌트의 실행에 필요한 컴포넌트들이 같이 선택된다. 예전에는 커널 컴포넌트들 간의 의존성과 응용 프로그램들 간의 의존성이 따로 관리되었으나 근래에는 타겟 이미지를 구성하는 모든 컴포넌트들의 의존성이 통합적으로 관리된다. 타겟 이미지 생성 도구는 다양한 타겟 보드에 대한 BSP(Board Support Package)를 제공하여 개발자가 빨리 타겟 이미지를 생성할 수 있게 한다. 근래에는 BSP보다 더 상위 수준인 임베디드 시스템의 응용 분야를 선택할 수 있다. 예를 들면, 개발자가 자동차 분야를 선택하면 이 분야의 임베디드 시스템이 공통적으로 필요로 하는 라이브러리와 유틸리티(예를 들면, CAN 지원 기능)를 포함한 타겟 이미지를 생성한다.

소프트웨어 개발 도구란 타겟에 내장되어 실행될 프로그램을 개발하는 데 필요한 도구다. 가장 기본적인 것으로서 크로스 컴파일 툴체인과 디버거가 있다. 그리고, 타겟 모니터링 및 프로파일링 도구, 디바이스 드라이버 개발 툴킷, GUI 빌더도 중요한 소프트웨어 개발 도구다. 더욱 편리한 개발을 위해서 위에서 언급한 개발 도구들이 통합개발환경(Integrated Development Environment)으로 제공된다.

설계 자동화 도구란 임베디드 소프트웨어를 개발할 때 시스템을 분석 설계하고 코드를 자동 생성한 후에 최종 시스템을 구축할 수 있도록 지원하는 기술로서 임베디드 소프트웨어 개발 프로세스 기술, 시스템 분석 및 모델링 기술, 소프트웨어

아키텍처를 기반으로 하는 임베디드 소프트웨어 개발 기술 등이 있다.

테스트 자동화 도구에는 화이트 박스, 블랙 박스, 회귀(regression), 유닛, 통합, 코드 범위, 오류 방지 테스트 등이 있다.

본 고에서는 두 번째인 임베디드 소프트웨어 개발 도구의 기술 및 시장 동향에 대해 기술한다.

3. 임베디드 소프트웨어 개발 도구 기술 동향

본 장에서는 임베디드 소프트웨어 개발 도구의 주요 기술들을 살펴본다. 이 기술들은 세계적인 임베디드 소프트웨어 개발 도구가 되기 위한 필요조건이라고 할 수 있을 것이다.

3.1 타겟 이미지 생성 도구와의 연동

개발을 완료했거나, 아니면 개발 중에도 프로그램을 타겟 이미지에 포함시켜서 프로그램이 정상적으로 타겟에서 실행되는지 테스트해 보아야 한다. 이 때 개발자가 프로그램을 매번 직접 타겟 이미지 생성 도구에서 선택될 수 있는 하나의 컴포넌트로 만들고, 이를 타겟 이미지 생성 도구에 추가한다면 매우 불편하다. 따라서, 임베디드 소프트웨어 개발 도구와 타겟 이미지 생성 도구가 연동되어서, 개발자가 개발한 프로그램이 자동적으로 하나의 컴포넌트로 만들어진 후 타겟 이미지 생성 도구에 추가될 수 있어야 한다.

3.2 통합개발환경

임베디드 소프트웨어를 편리하게 개발하기 위해서는 뛰어난 통합개발환경이 필수적이다. 대표적인 통합개발환경은 다음과 같은 기능을 갖고 있다.

- 다양한 호스트 플랫폼 지원
- 다양한 타겟 지원
- 프로젝트 관리자

- 빌드, 디버깅, 실행 설정 관리
- 편리한 소스 네비게이션
- 버전 제어

통합개발환경은 점차 다양한 호스트 플랫폼을 지원하는 추세다. 개발자가 어떤 플랫폼에 있더라도 사용하던 익숙한 통합개발환경을 이용할 수 있어야 한다. 임베디드 리눅스용 통합개발환경인 TimeSys의 TimeStorm이나 QNX의 Neutrino RTOS 용 통합개발 환경인 Momentics는 윈도우즈, 리눅스, 솔라리스 등의 호스트 플랫폼을 지원하고 있다.

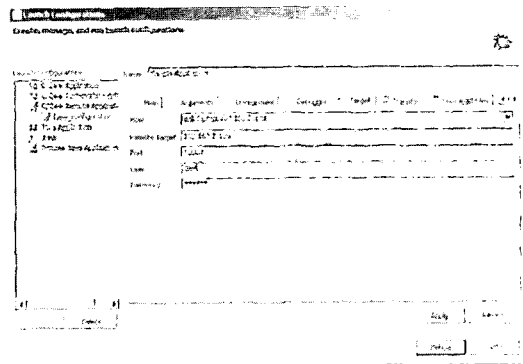
통합개발환경은 다양한 타겟에서 실행될 수 있는 프로그램의 개발을 지원한다. 이를 위해서는 통합개발환경의 크로스 컴파일러, 크로스 유틸리티, 크로스 디버거가 X86, PowerPC, ARM, StrongARM, XScale, MIPS, SH 등의 다양한 타겟을 지원해야 한다.

통합개발환경은 프로젝트 관리자를 지원한다. 프로젝트 관리자는 개발자가 일일이 Makefile을 만들고 유지보수하는 수고를 없애준다. 프로젝트 관리자는 빌드 설정을 관리한다. 개발자가 일일이 타겟에 맞는 크로스컴파일러와 링커의 위치를 지정하고 응용에 필요한 라이브러리를 설정할 필요없이, 타겟만 선택하면 해당하는 크로스컴파일러와 링커를 자동으로 선택할 수 있거나 응용의 종류를 선택하면 필요한 라이브러리도 자동으로 선택된다. 임베디드 리눅스용 응용을 개발할 때에는 공개된 소스 코드를 이용하는 경우가 많은데 이들은 대부분 automake와 autoconf에 의해서 Makefile이 생성된다. 공개 소스 코드를 통합개발환경의 프로젝트로 개발자가 직접 변환하는 것은 매우 불편하다. 따라서, 프로젝트 관리자는 Makefile 기반의 공개 소스 코드를 가능하면 편리하게 프로젝트로 변환해주는 기능도 필요하다.

임베디드 소프트웨어를 위한 프로젝트 관리자는 빌드 설정 외에도 디버깅과 실행에 관련된 설정도

필요하다. 현재 개발중인 프로그램을 어떤 타겟에 전송하여 실행시킬 것인지와, 타겟에서 프로그램을 실행시킬 때 필요한 옵션, 또는 디버깅할 때의 옵션 등을 설정한다. 이 프로그램의 실행에 필요한 라이브러리나 데이터 파일을 설정하면 프로그램과 함께 타겟으로 전송된다. 개발자는 한 번만 이와 같은 설정을 해두면, 원격 실행 버튼만 클릭함으로써 간단하게 호스트에서 개발 중인 프로그램을 타겟에 전송하고 실행시켜서 테스트해 볼 수 있다. 또한, 원격 디버깅 버튼만 클릭함으로써 타겟으로 프로그램을 전송하고 원격 디버깅을 시작할 수 있다. 임베디드 소프트웨어 개발 시에 원격 실행과 원격 디버깅은 매우 빈번하게 발생하는 작업이므로 개발자가 이를 편리하게 할 수 있어야 한다.

(그림 1)은 TimeSys 사의 임베디드 소프트웨어 통합개발환경인 TimeStorm에 있는 Launch Configurations 윈도우즈인데, 타겟의 IP 주소, 포트 등의 정보를 설정한 것을 보여준다. 이 외에도 타겟에서 실행시킬 프로그램의 인자와 환경변수, 디버깅 옵션, 타겟에 전송할 파일들을 설정할 수 있다.



(그림 1) TimeStorm의 Launch Configurations 화면

통합개발환경은 소스 코드를 편리하게 네비게이션할 수 있도록 지원한다. 원하는 함수의 선언이나 정의로 빨리 찾아가거나, 클래스를 브라우징하는

것, 함수의 호출 관계를 보여주는 것 등 소스코드에서 원하는 부분을 빨리 찾아갈 수 있도록 해주는 기능은 기존 통합개발환경에 이미 있는 것들이지만, 임베디드 소프트웨어용 통합개발환경은 이런 기능들이 부족하였다. 최근 임베디드 소프트웨어용 통합개발환경들은 점차 편리한 소스 네비게이션 기능을 지원하고 있다.

통합개발환경은 버전 관리를 지원한다. 임베디드 소프트웨어도 점점 복잡하고 커짐에 따라 소스 코드의 버전 관리가 필요하다.

3.3 디버깅

임베디드 소프트웨어 개발이란 응용 프로그램 개발뿐만 아니라 새로운 타겟 하드웨어로 커널 포팅, 커널에 새로운 기능 추가, 커널의 기능 수정, 그리고 다바이스 드라이버 개발도 포함한다. 이를 위하여 커널을 디버깅할 수 있는 디버거가 필요한데, 소프트웨어적인 방법과 하드웨어를 이용하는 방법이 있다. KGDB는 리눅스 커널을 디버깅할 수 있게 해주는 프로그램이다. 그런데, 커널이 죽으면 호스트의 디버거와 타겟 간의 통신 채널도 죽게 되어 원인을 파악하기 어렵다.

좀 더 신뢰성 있는 디버깅을 하기 위해서는 JTAG/BDM을 지원하는 하드웨어를 이용하여 디버깅해야 한다. 그러면, 운영체제가 죽더라도 어느 위치에서 어떤 상태로 죽었는지를 알아낼 수도 있고, 가상 메모리와 물리 메모리(Physical Memory)의 상태를 알 수 있다. JTAG/BDM 하드웨어를 이용할 경우, 커널 디버깅 외에도 커널을 새로운 하드웨어로 포팅하는데 필요한 몇 가지 중요한 작업을 할 수 있다. 타겟에 부트로더가 없는 경우에도 플래시 메모리 퓨징(fusing) 및 이미지 다운로드를 할 수 있고, 부트로더를 포팅하거나 개발 및 디버깅하는데 JTAG/BDM 하드웨어가 이용된다. 대부분의 개발 도구 제공 업체는 JTAG/BDM 하드웨

어 솔루션을 제공한다. 예를 들면, TimeSys는 자사의 TimeStorm 통합개발환경과 함께 사용할 수 있는 BDI2000이라는 JTAG 장비를 제공하고 Wind River는 VisionProbe II를 제공한다.

3.4 모니터링 및 프로파일링

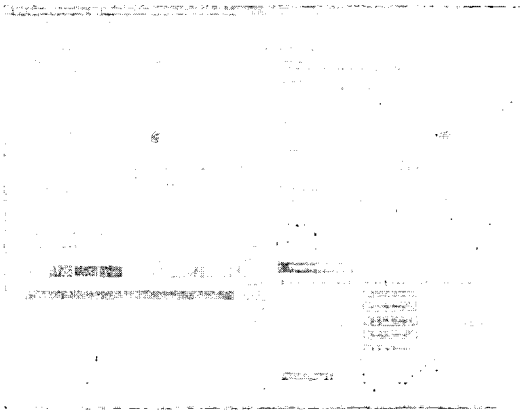
일반적인 소프트웨어는 단지 원하는 기능이 정상적으로 실행되면 된다. 그러나, 임베디드 소프트웨어의 경우, 이것이 타겟에서 실행되면서 CPU나 메모리와 같은 자원을 얼마나 소비하고 어느 정도 성능을 발휘하는지 파악하는 것이 매우 중요하다. 또한, 프로그램이 실행되면서 발생하는 상세한 커널 이벤트의 분석도 필요하다.

타겟 모니터는 타겟에서 실행되는 프로세스나 스레드의 리스트, 각 프로세스의 CPU, 메모리(스택, 힙, 텍스트) 사용량, 타겟 전체의 메모리 사용량 등을 보여준다. 타겟 모니터는 이 정보들을 개발자가 한 눈에 파악하기 용이하도록 그래프로 표현하기도 한다.

타겟 프로파일러는 실시간으로 타겟에서 발생하는 이벤트를 보여주거나, 실행시에는 이벤트를 로깅만 하고 나중에 이들에 대한 정보를 보여줄 수도 있다. 전자를 실시간 프로파일링이라 하고 후자를 사후(Post-Mortem) 프로파일링이라고 한다. 이미 실행 중인 프로세스나 태스크에 프로파일러를 연결하여 프로파일링을 하기도 한다(동적 프로파일링). 함수나 소스코드 라인 별로 CPU 사용시간을 측정하기도 하며, 각 태스크나 프로세스가 어떻게 스케줄링되고 있는지도 그래프로 보여준다. 프로파일링을 하는 대상이 너무 많으면 타겟 시스템에 부하를 줄 수도 있고, 정작 원하는 정보를 얻기 어렵기 때문에 이벤트 필터링 기능을 제공한다. 또한 개발자가 새로운 이벤트를 정의하여 시스템 이벤트들과 함께 프로파일링할 수도 있다.

타겟 모니터와 타겟 프로파일러도 통합개발환경

안에 통합되고 있다. 통합된 타겟 모니터는 디버거와 같은 도구와 연동되기도 한다. 예를 들면, 타겟 모니터의 프로세스 리스트에서 하나의 프로세스를 선택한 후 이 프로세스에게 원하는 시그널을 보내거나, 이 프로세스를 디버거에 연결시켜서 동적으로 디버깅을 시작할 수 있다. (그림 2)는 QNX Momentics의 타겟 시스템 모니터 화면을 보여준다.

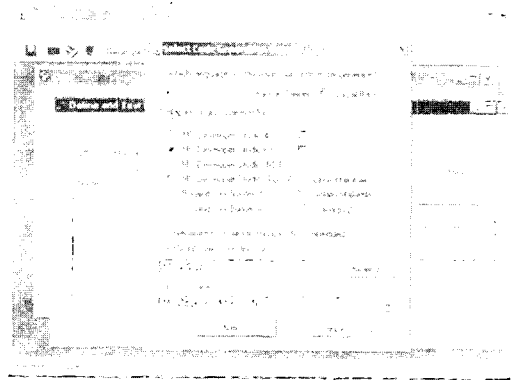


(그림 2) QNX Momentics의 타겟 시스템 모니터링 화면

3.5 디바이스 드라이버 개발 툴킷

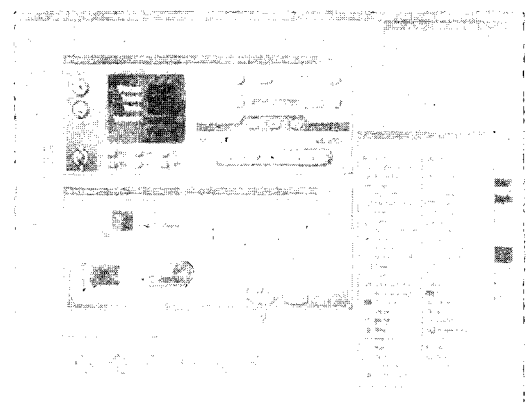
디바이스 드라이버 개발 툴킷은 커널이나 디바이스에 대한 깊은 지식 없이도 디바이스 드라이버를 빠른 시간 안에 개발할 수 있게 해준다. 디바이스의 타입만 선택하면 디바이스가 사용하는 리소스와 상태를 테스트하여 알려주고, 이를 바탕으로 원하는 디바이스 드라이버 소스코드를 자동 생성한다. 개발자는 디바이스 드라이버의 특수한 기능만 구현하고 테스트하면 되므로 개발 기간을 단축할 수 있다. 대표적인 디바이스 드라이버 개발 툴킷으로 CompuWare의 Driver Studio[8], MS의 DDK[9], Jungo의 WinDriver[7]가 있다. 이 중에서 WinDriver는 위에서 언급한 기능을 GUI 기반의 편리한 워저드로 제공하며, 다양한 운영체제(윈도우즈 계열, 리눅스, 솔라리스, VxWorks)용 디바이스 드라이버 개발을 지원한다. (그림 3)은 Jungo의

WinDriver 디바이스 드라이버 개발 툴킷에서 디바이스 드라이버 하드웨어의 특징을 선택한 후에 원하는 운영체제용 소스 코드를 생성하기 위하여 개발환경을 선택하는 단계를 보여준다. 디바이스 드라이버 소스 코드의 프로그래밍 언어와 운영체제에 따라서 다양한 프로젝트나 Makefile의 종류를 선택할 수도 있다.



(그림 3) Jungo의 WinDriver 디바이스 드라이버 개발 툴킷

3.6 GUI 빌더



(그림 4) QNX의 GUI 빌더, PhAB

과거에는 흑백의 단순한 텍스트만을 보여주던 임베디드 시스템들이 점차 화려하고 멋진 그래픽을 보여주는 디스플레이를 탑재하고 있다. 요즘 생

산되는 핸드폰들은 모두 컬러 LCD를 탑재하고 있고, 320X240 이상의 해상도를 가진 컬러 LCD를 탑재한 PDA의 사용이 늘어나고 있다. 임베디드 GUI 응용 프로그램을 빨리 개발할 수 있도록 임베디드 소프트웨어용 통합개발환경도 GUI 빌더를 제공해야 한다. QNX는 Photon microGUI라는 임베디드 소프트웨어용 그래픽 윈도우 시스템을 제공하는데, 이를 위한 GUI 빌더로 PhAB(Photon Application Builder)를 제공한다.

4. 임베디드 소프트웨어 개발 도구 시장 동향

4.1 임베디드 소프트웨어용 개발 도구의 동향

임베디드 소프트웨어 개발 도구의 시장 규모는 2002년의 전체 임베디드 소프트웨어 시장 중에서 약 24%를 차지한다. 세계적인 경기 침체로 인하여 2002년의 시장규모는 2001년의 시장규모에 비해 감소되었지만, 임베디드 시스템 제조업자들은 프로젝트 개발 기간을 단축하기 위해 편리하고 기능이 풍부한 개발 도구를 매우 원하고 있으므로 향후 임베디드 소프트웨어 개발 도구 분야의 전망은 매우 밝다. 이에 따라, 연평균 9.5%씩 시장 규모가 성장하여 2007년에는 5억 달러를 넘을 것이다[1]. 이 시장 규모는 단품으로 판매되는 임베디드 소프트웨어 개발 도구만 고려한 것이고, 하드웨어나 운영체제와 함께 번들로 제공된 임베디드 소프트웨어 개발 도구는 제외된 것이다.

최근 임베디드 소프트웨어 개발 도구는 단품으로 판매되는 경우보다 하드웨어나 운영체제와 함께 번들로 제공되는 예가 증가하고 있다. 전체 개발 프로세스에서 필요한 운영체제와 도구, 그리고 기술지원을 함께 제공하거나, 특정 응용 분야에 필요한 미들웨어, 프로토콜 스택 등을 함께 제공함으로써 개발자가 더욱 빨리 임베디드 소프트웨어를

개발할 수 있게 한다. 예를 들면, MontaVista는 Carrier Grade 에디션과 Consumer Electronics 에디션을 제공하는데, 전자는고가용성(high availability)이 필요한 통신장비에 적합한 임베디드 리눅스와 미들웨어, 그리고 개발 도구로 구성되고, 후자는 가전제품에 맞도록 저전력 관리 등의 기능이 특화된 임베디드 리눅스와 미들웨어, 그리고 개발 도구로 구성된다[11]. 점점 더 많은 임베디드 소프트웨어 벤더들이 이와 같은 방식으로 자신들의 솔루션을 판매하고 있으므로 실제 임베디드 소프트웨어 개발 도구의 시장 규모는 <표 1>의 시장 규모보다 더 클 것이다.

<표 1> 단품으로 판매되는 임베디드 소프트웨어 개발 도구의 세계 시장 규모

<단위:백만달러>

2001년	2002년	2003년	2004년	2005년	2006년	2007년	연평균성장률
376.1	333.1	351.1	323	425	471	525.2	9.5%

<출처>VDC, 2002

임베디드 소프트웨어 개발 도구에서 디버거나 컴파일러만을 판매하는 경우보다 통합개발환경(IDE)으로 구매하는 경우가 점차 증가하고 있다. 임베디드 소프트웨어 개발 도구에서 통합개발환경이 차지하는 비중이 2002년에 40.6%에서 2007년에는 44.6%로 증가할 것이다. 단품으로 판매되는 컴파일러나 디버거의 시장 규모는 2007년까지 연평균 성장률이 각각 5.8%와 5.5%이지만, 통합개발환경의 시장 규모는 11.6%의 높은 연평균 성장률을 보여줄 것으로 예측된다.

<표 2> 통합개발환경, 컴파일러, 디버거의 세계시장 규모

<단위:백만달러>

	2002년	2007년	연평균 성장률
통합개발환경	135.2(40.6%)	234.5(44.6%)	11.6%
컴파일러	37.6(11.3%)	49.9(9.5%)	5.8%
디버거	39.0(11.7%)	50.9(9.7%)	5.5%

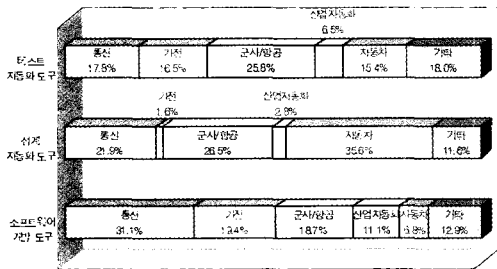
<출처>VDC, 2002

임베디드 소프트웨어 개발 도구가 주로 사용되는 응용 분야는 통신(Telecommunication), 가전(Consumer Electronics), 군사/항공(Military/Aerospace), 산업자동화(Industrial Automation), 자동차(Automotive)의 다섯 가지 분야다. 시장 규모는 통신, 가전, 군사/항공, 산업자동화, 자동차 순인데, 이 중에서 가전 분야와 군사/항공 분야가 큰 폭으로 성장할 것이다.

<표 3> 응용 분야별 소프트웨어 개발 도구 세계 시장 규모
<단위:백만달러>

	2002년	2007년	연평균 성장률
통신	103.6(31.1%)	147.8(28.1%)	7.4%
가전	64.6(19.4%)	112.2(21.4%)	11.7%
군사/항공	62.3(18.7%)	108.1(20.6%)	11.7%
산업자동화	37.0(11.1%)	55.4(10.5%)	8.4%
자동차	22.7(6.8%)	37.2(7.1%)	10.4%
기타	42.9(12.9%)	64.5(12.3%)	8.6%

<출처> IDC, 2002



(그림 5) 소프트웨어 개발 도구, 설계자동화 도구, 테스트 자동화 도구의 응용 분야별 시장 점유율 비교(2002년)

임베디드 소프트웨어 개발 도구와 설계 자동화 도구, 그리고 테스트 자동화 도구는 응용 분야별로 주요 시장의 규모가 다르다. (그림 5)는 세 가지 분야의 도구별로 어떤 응용 분야에서 수요가 많은지 보여준다. 소프트웨어 개발 도구는 통신, 가전, 군사/항공 순이고, 설계 자동화 도구는 자동차, 군사/항공, 통신 분야 순으로, 그리고 테스트 자동화 도구는 군사/항공, 통신, 가전 분야 순으로 많이 사용되고 있다. (그림 5)를 통해서 응용 분야별로 어떤

종류의 도구가 중요한지도 추측해 볼 수 있다. 자동차 분야와 군사/항공 분야에서는 상대적으로 설계 및 테스트 자동화 도구가 매우 중요한 반면 가전 분야에서는 그렇지 않음을 알 수 있다.

4.2 임베디드 리눅스 분야에서 개발 도구의 동향

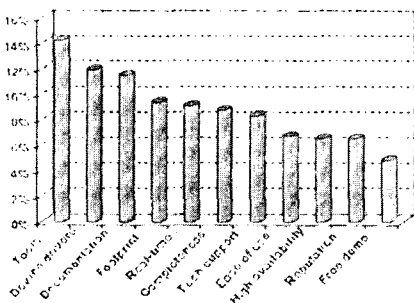
임베디드 운영체제 시장에서 임베디드 리눅스의 비중은 점차 커지고 있다. 2003년 6월에 마쓰시타, 소니, 히타치, NEC, 필립스, 샤프, 도시바, 삼성이 가전기기에 적합한 리눅스를 개발하기 위한 CE 리눅스 포럼(Consumer Electronics Linux Forum)을 만들었고 그 후로 LG, IBM, 노키아, TimeSys, MontaVista 등의 유명 업체들이 이 포럼에 참여하였다.

현재 임베디드 소프트웨어 업계에서 시장점유율 1위인 Wind River 사는 그동안 리눅스에 비판적이었으나, 최근 임베디드 리눅스를 지원하는 개발 도구를 출시하였다. 2003년 10월에 자사의 온칩 디버깅(On-Chip Debugging) 솔루션인 visionPROBE II가 임베디드 리눅스를 지원한다고 발표하였다. 앞으로 Wind River사의 더 많은 개발 도구들이 임베디드 리눅스를 지원할 것으로 예측된다. Wind River의 RTOS인 VxWorks 사용자들 중에서도 임베디드 리눅스를 동시에 사용하는 개발자들이 늘어나면서, Wind River사의 좋은 도구들을 임베디드 리눅스에서도 사용하기를 원했기 때문에 Wind River사가 입장을 바꾼 것으로 분석가들은 해석하고 있다.

EDC에서 2003년 3월에 발표한 자료[6]에 의하면 현재 또는 다음 임베디드 시스템 프로젝트에서 리눅스를 사용할 것이라는 응답이 50%가 넘었다. 반면 20%의 개발자만이 임베디드 리눅스의 소프트웨어 개발 도구가 좋다고 했으며 5%만이 아주 좋다고 했다. 60% 이상의 개발자는 보통이나 좋지 않다고 응답하였다.

이처럼 임베디드 리눅스를 사용하는 예가 급증하는 반면, 임베디드 리눅스용 소프트웨어 개발 도구는 다른 임베디드 운영체제용 개발 도구에 비해서 빈약하기 때문에 임베디드 리눅스용 개발 도구의 필요성이 더욱 대두되고 있다. Linux Devices.com에서 2003년 5월에 발표한 자료[5]에 의하면 임베디드 리눅스의 벤더나 소스를 선택하는데 있어서 가장 중요하게 생각하는 것은 개발 도구였다.

국내 시장의 경우 임베디드 소프트웨어 개발환경은 GNU 개발 도구들을 개별적으로 지원하는 형태다. 한국전자통신연구원에서는 임베디드 리눅스인 Qplus용 통합개발환경으로 Esto[10]를 개발하였다. Esto 초기 버전은 국내 업체에 기술이전되어 상용화 준비 단계에 있다.



(그림 6) 임베디드 리눅스벤더의 선택에 영향을 미치는 것[5]

4.3 이클립스(Eclipse) 프로젝트

현재, 그리고 앞으로 소프트웨어 개발 환경 시장에 큰 영향을 미칠 요소로 이클립스 프로젝트가 있다. 이클립스 프로젝트의 목표는 다양한 개발 도구들이 쉽게 통합될 수 있는 통합개발환경을 개발하는 것이다. 이클립스는 1999년에 IBM의 연구원들에 의해 개발되기 시작하였는데, 2001년에는 공개 소스 프로젝트가 되고 이클립스 컨소시엄 (<http://eclipse.org>)이 구성되었다. 이클립스의 초

기 멤버로는 Borland, MERANT, QNX, Ratical, RedHat, SuSE, TogetherSoft가 있고, 그 후에 Fujitsu, Sybase, Oracle, OMG, MontaVista, TimeSys, Intel, Ericsson 등의 유명한 업체들이 참여하고 있다. 한국전자통신연구원도 2002년 8월부터 이클립스 멤버로 활동하고 있다.

이클립스에 어떤 매력이 있어서 세계적으로 유명한 업체들이 참여하고 있는 것인지 예를 들어서 설명한다. 개발자는 어떤 프로젝트를 수행하면서 다양한 개발 도구를 사용하게 된다. 시스템을 모델링하는데 도구 A를 사용한 후, 이를 구현하기 위하여 Z라는 프로그래밍 언어를 지원하는 통합개발환경으로 B 제품을 사용하고 있다. 뿐만 아니라, 프로그램의 모니터링과 프로파일링을 하기 위해서 C라는 도구, 테스트를 위하여 D라는 도구, 소스 코드 버전 제어를 위하여 E라는 도구를 사용한다. 개발을 마친 후에는 문서화를 위하여 도구 F를 사용한다. 이 개발자는 다른 프로젝트에서 Y라는 프로그래밍 언어를 사용해야 하기 때문에 통합개발환경으로 G 제품을 사용한다. 이처럼 한 개발자는 하는 일에 따라서 별개의 도구를 사용하는 방법을 익혀야 하고, 이 도구들간의 데이터가 서로 호환되지 않아서 개발자가 직접 변환해야 할 수도 있다. 만약, 이와 같은 다양한 도구들이 하나의 플랫폼에 유기적으로 통합된다면 개발자는 무척 편리하지 않을까. 이것이 이클립스가 등장하게 된 배경이자 목표다.

이클립스 플랫폼은 공개 소스 도구들을 유기적으로 통합하는 플랫폼으로써 개발자에게 확장성과 일관성을 제공한다. 개발자는 원하는 도구를 자신의 이클립스 플랫폼에 플러그인하면 되므로 새로운 도구를 쉽게 추가하여 확장할 수 있다. 이클립스에 플러그인 되는 도구로 만들 수 있는 응용에도 제한이 없어서 개발자는 다양한 언어(C, C++, Java, Pascal, HTML, UML 등)를 지원하는 도구

를 이클립스에 플러그인하여 사용할 수 있다. 또한, 원하는 회사의 컴파일러나 디버거, 또는 웹에디터를 구입하거나 공개 소스를 가져와서 사용할 수 있다. 이클립스에 플러그인되는 도구는 일관된 모습 (look & feel)을 제공하므로 플러그인 된 새로운 도구의 사용 방법을 쉽게 익힐 수 있다. 이클립스는 리눅스, 유닉스, 윈도우즈, MAC OS X 등의 다양한 플랫폼에서 실행된다.

이클립스는 공개 소스 커뮤니티다. 따라서, 도구 회사 측면에서는 이클립스 프로젝트에 참여함으로써 서로 기술을 공유할 수 있으며 자신만의 장점을 살려서 더욱 새로운 도구 개발에만 집중할 수도 있다. <http://eclipse.org>에서는 기술 공유를 위하여 여러 가지 부 프로젝트들이 진행되고 있다.

이처럼 이클립스는 개발자와 도구 회사에게 많은 장점이 있기 때문에 유명한 회사들이 이클립스 프로젝트 멤버로 직접 참여하여 활동하고 있고, 멤버들은 꾸준히 증가하고 있다. 이클립스 기반 임베디드 소프트웨어 통합개발환경으로는 현재 TimeSys의 TimeStorm과 Montavista의 DevRocket, 그리고 QNX의 Momentics가 대표적이다. TimeStorm과 DevRocket은 임베디드 리눅스용 통합개발환경이고, 후자는 QNX의 Neutrino 운영체제용 통합개발환경이다.

5. 결 론

앞으로 임베디드 시스템의 응용 분야는 더욱 다양해지고 그 수요도 지속적으로 증가할 것이다. 이에 따라 임베디드 시스템을 적시에 개발하기 위하여 더욱 편리하고 기능이 풍부한 임베디드 소프트웨어용 개발 도구의 중요성이 대두된다. 특히, 임베디드 리눅스는 앞으로 가장 널리 쓰이게 될 임베디드 운영체제임에도 불구하고 만족할만한 개발 도구가 부족해서 개발이 시급한 분야다.

국내에서는 2002년도에 한국전자통신연구원

에서 임베디드 리눅스인 Qplus용 통합개발환경으로 Esto를 개발하여 발표하였다. 2004년 2월에는 기능이 더욱 강화된 이클립스 기반의 Esto가 이클립스 컨퍼런스에서 발표되어 저전력 지원 도구, 디바이스 드라이버 개발 도구 등의 차별화된 기능들에 대해서 호평을 받은 바 있다. Esto는 임베디드 리눅스 뿐만 아니라 한국전자통신연구원에서 개발할 예정인 마이크로 운영체제와 나노 운영체제도 지원할 예정이다. 마이크로 운영체제는 Wind River의 VxWorks처럼 유저 모드와 커널 모드의 구분없이 단일한 주소 공간을 제공하고 경성 실시간 시스템을 지원한다. 나노 운영체제는 UC Berkeley의 TinyOS처럼 작은 센서에도 사용될 수 있는 초소형 운영체제다.

임베디드 소프트웨어는 크기도 작아야 하지만, 전력 소모도 적어야 한다. 이를 위하여 개발 중인 임베디드 소프트웨어가 소모하는 전력을 측정해주는 도구가 Esto에 포함되어 있다. 실시간 프로그램의 경우 기존 디버거에서 하듯이 정지점에서 멈추게 되면 제대로 디버깅을 하기 어렵다. 이를 위하여 비정지 디버깅(Non-Stop Debugging 또는 Tracepoint Debugging)을 지원하는 디버거도 Esto에 포함되어 있다. 이 외에도 경성 실시간 시스템의 개발에 필요한 실시간성 분석 도구가 개발 중에 있고, 향후에는 유비쿼터스 환경의 임베디드 소프트웨어의 개발을 도와주는 도구들을 추가 개발해 나아갈 것이다.

참고문헌

- [1] "The Embedded Software Strategic Market Intelligence Program 2002/2003," VDC (Venture Development Corporation), Feb., 2003.
- [2] S. Shavor, J. D'anjou, S. Fairbrother, etc., 'The Java Developer's Guide to Eclipse.'

Addison Wesley, 2003.

- [3] 윤성준, 조상민, 송정일, 'Java 세상을 덮치는 Eclipse,' 인사이트, 2003..
- [4] <http://www.eclipse.org>
- [5] Rick Lehrbaum, "Snapshot of the Embedded Linux market," LinuxDevices.com, May, 2003, <http://www.linuxdevices.com/articles/AT7301151332.html>.
- [6] Rick Lehrbaum, "EDC: Embedded Linux remains #1 choice of developers - despite tools dissatisfaction," <http://www.linuxdevices.com/articles/AT4787985721.html>.
- [7] <http://www.jungo.com/windriver.html>
- [8] <http://www.compuware.com/products/driverstudio/default.htm>
- [9] <http://www.microsoft.com/whdc/ddk/winddk.mspx>
- [10] <http://esto.etri.re.kr>
- [11] 몬타비스타 소프트웨어 코리아(주), "몬타비스타, 임베디드용 리눅스 제품군," Real-Time Embedded World, Vol.9, Sep. 2003, pp.110-114.

저자약력



임 채 덕

1989년 전남대학교 전산통계학과 학사
 1999년 충남대학교 전산학 석사
 현재 충남대학교 전산학 박사과정 재학
 1989년 한국전자통신연구원 연구원
 현재 한국전자통신연구원
 임베디드S/W개발도구 연구팀장
 관심분야 : 임베디드S/W개발도구 및 시스템 S/W,
 실시간컴퓨팅 및 미들웨어