



# 임베디드 S/W 개발도구 기술

(주)코스모 대표이사 조 주 현



## ▷▷▷ 임베디드 S/W 특집

- 임베디드 S/W 산업 육성 추진방안
- 임베디드 S/W 산업 동향 및 향후 전망
- IP 셋톱박스 S/W 플랫폼 기술
- 스마트 센서 노드 운영체제 기술
- **임베디드 S/W 개발도구 기술**
- SCM기반의 임베디드 S/W 인력양성 정책

## 요약

초고속인터넷, 무선통신의 급속한 보급과 유비쿼터스 네트워크의 세계적인 추세에 따라 그 기반 기술이 되는 임베디드 시스템이라는 용어가 과거 통신 및 제어장치에서만 사용되었으나 이제 일상생활의 필수품이 되고 전 산업에 영향을 주게 되었다. 통신시스템, 제어장치, 휴대폰, 디지털 TV, 셋톱박스, 홈네트워크에서부터 스마트카드, RFID, 센서노드 등 임베디드 소프트웨어가 들어가지 않는 곳이 없을 정도로 광범위하며 단순한 제어에서부터 복잡한 프로토콜 스택 및 기능이 다양하다. 반면에 임베디드 소프트웨어의 개발 환경이 일반 소프트웨어의 개발환경보다 훨씬 어려워서 효과적인 개발도구가 없이는 경쟁력 있는 개발이 불가능하다. 결국 임베디드 소프트웨어의 개발 경쟁력이 국가의 경쟁력이라 해도 과언이 아니므로 효과적인 임베디드 소프트웨어 개발환경의 구축은 매우 중요하다. 이러한 임베디드 소프트웨어 개발도구의 국내외 개발현황과 기술수준을 소개한다. 특히, 국내에서 세계적 수준의 ESTO(Embedded Software Toolkit)의 상용화는 우리나라에서 특히 취약한 핵심 시스템 소프트웨어 분야에서 처음으로 세계시장에서 경쟁력을 자랑하게 된 점이 주목할 만하다.

## I. 서론

최근 초고속인터넷 및 무선 통신환경의 급속한 보급과 더불어 디지털 TV, 인터넷 셋톱박스, 개인 휴대통신 단말기 및 각종 정보형 단말기 등이 일상생활에 급속히 파고들며 따라 종래 통신장치나 자동제어 등의 제한된 영역에서나 들어 볼 수 있었던 임베디드 시스템이라는 용어가 이제는 일상 생활에서 흔히 접해볼 수 있게 되었다. 아울러 임베디드 시스템 시장은 급격히 팽창하고 있으며 데스크탑 PC시장을 추월할 것으로 예상되고 있다.

그러나 임베디드 시스템은 일반 PC환경과 달리 하드웨어 플랫폼 종류와 규모의 다양성, 개발시스템(Host)과 목적시스템(Target)이 다른 교차개발환경 등의 특수성으로 인하여 시스템 기능이 구현되는 소프트웨어의 개발 특히 프로그램의 시험이 일반 PC환경보다 몇 배로 어렵다. 이러한 점으로 인하여 임베디드 시스템의 납기를 놓치거나 실패하는 경우가 많다.

따라서 임베디드 시스템은 신뢰성 높은 검증된 실시간 운영체제와 그 위에서 동작하는 풍부한 라이브러리 및 응용소프트웨어의 지원이 중요하다. 그러나 무엇보다 더 중요한 것은 교차개발환경에서 소프트웨어 시험을 효과적으로 가능하게 해주는 GUI기반의 편리한 교차디버깅도구가 제공되지 않으면 안된다. 국내에서도 실시간 운영체제의 개발 사례는 있으나 이러한 종합적인 환경이 뒷받침되지 않으면 첨단 외국 제품에 비해 아무런 경쟁력을 갖지 못한다.

임베디드 시스템용 실시간 운영체제와 소프트웨어 개발환경 분야에서 대표적인 외국제품은 세계시장의 대부분을 점하고 있는 미국 Wind River사의 Tonado/VxWorks이 있다. 국내에서도 아직은 이러한 외국 제품들을 전량 수입하여 사용하고 있는 실정이다. 이러한 외국 제품은 풍부한 라이브러리, 프로토콜 및 응용 소프트웨어 뿐만 아니라 다양한 디바이스 드라이버 및 보드 지원 패키지(Board Support Package 혹은 Hardware Abstraction Layer)의 지원에 의한 높은

하드웨어 적응성 등의 장점 때문에 많이 사용되고 있다. 그러나 이러한 외국 제품은 신속한 기술지원의 어려움 뿐만 아니라 고가의 개발환경의 구입비용과 최종 임베디드 제품 단위별 로열티(run-time royalty)의 부담 때문에 저가의 임베디드 시스템의 개발에 널리 사용되지 못하고 있는 실정이다. Windows-CE의 경우에는 정보형 단말기용으로 사용자 인터페이스 개발환경은 매우 좋으나 저가 단말기의 적은 마진에 비해 로열티가 부담이 되어 개발은 해놓고 납품을 할 수 없는 경우가 많다[1].

임베디드 시스템 소프트웨어는 크게 타겟에서 동작하는 런타임 솔루션 소프트웨어와 개발용 호스트 시스템에서 이루어지는 개발도구가 있다. 이 논문에서는 임베디드 시스템 소프트웨어의 주요 특징을 다시 살펴본 다음에 개발도구의 세계적 현황과 국내 개발현황을 소개한다.

## II. 임베디드 소프트웨어의 특징

임베디드 시스템은 과거와는 달리 이제 복잡한 기능이 요구되므로 단순한 제어 프로그램만으로는 실현이 불가능하고 실시간 운영체제 나아가 임베디드 리눅스 등 본격적인 임베디드 운영체제를 기반으로 대형의 소프트웨어를 개발해야 가능하게 되었다. 그럼에도 단말기 등 많은 저가의 임베디드 시스템은 프로세서 및 메모리 등의 한계성으로 인하여 소프트웨어는 소형이면서 고속처리가 가능해야 한다. 또한 임베디드 시스템은 일반 PC시장과 달리 하드웨어 플랫폼의 종류가 매우 다양할 뿐만 아니라 그 응용 분야도 초소형 제어에서부터 대형 제어에 이르기까지 매우 다양하여 표준의 단일 솔루션을 제공하기가 매우 어렵다고 할 수 있다. 더욱이 임베디드 프로세서는 발전 속도가 매우 빠르므로 임베디드 프로세서도 쉽게 변경될 수 있어야 한다.

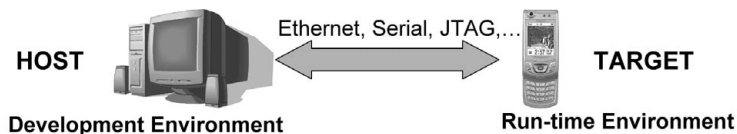


그림 1. 임베디드 소프트웨어의 개발 특성

특히, 임베디드 시스템은 소프트웨어 개발환경에서 일반 시스템과 크게 다르다. 즉 PC등 소프트웨어개발용 일반 컴퓨터 시스템(Host System)과 개발된 소프트웨어가 실제로 수행하는 목적 시스템(Target System)이 분리된 교차개발(Cross Development)환경이라는 것이다. 따라서 소프트웨어의 개발 특히 시험이 매우 어려우므로 이에 대한 효과적인 개발도구를 제공하는 것이 임베디드 시스템의 가장 큰 과제라 할 수 있다.

따라서 임베디드 시스템의 개발 환경은 그 구성요소인 임베디드 프로세서, 실시간 운영체제, 교차 디버거 등 개발도구, 그래픽 라이브러리, 임베디드 응용 소프트웨어 등은 서로 다른 종류가 상호 연동이 될 수 있는 개방형 구조가 되어야 하며 또한 이식성이 매우 높은 구조로 되어야 한다. 실시간 운영체제는 하부의 머신인 하드웨어가 바뀌어도 쉽게 이식할 수 있는 구조로 되어야 하며 위로는 응용소프트웨어와의 인터페이스(API)가 표준을 가능한 한 따르면서 간단하여야 한다. 그러므로서 다른 종류의 실시간 운영체제 혹은 Linux 등에서 동작하는 라이브러리, 네트워크 프로토콜 및 응용소프트웨어 등이 쉽게 이식될 수 있어야 한다.

교차개발도구는 타겟 임베디드 프로세서 및 운영체제가 달라도 쉽게 지원이 가능해야 한다. 과거에는 타겟 운영체제와 개발도구가 결합되어 분리할 수 없는 한 세트 지원되었으나 다양한 사용자의 요구를 충족하기 위하여서는 이들도 분리할 필요가 있다. 또한 개발도구의 사용자 인터페이스는 표준이 없으나 교차개발환경에 고유한 새로운 인터페이스를 설계하기보다 기존 PC환경 등에서 프로그램 개발에 많이 사용되는 인터페이스를 기본으로 하여 임베디드 실시간 디버깅에 고유한 것을 추가 및 보완함으로써 가능한 한 교차개발환경을 몰

라도 교차개발환경에서 프로그램 디버깅을 손쉽게 할 수 있는 사용자 인터페이스 투명성(transparency)을 제공하는 것이 교차개발도구의 목적이라 할 수 있다.

특정 임베디드 시스템의 사용자 인터페이스를 설계하기 위한 그래픽 라이브러리, 외부 시스템과의 통신을 위한 통신프로토콜 등의 라이브러리, 기타 모든 임베디드 시스템 소프트웨어는 가능한 한 표준을 따르면서도 임베디드 시스템에 실장되기 위해서는 그 크기가 소형이어야한다[1].

### III. 임베디드 S/W 개발도구의 세계적 현황

임베디드 소프트웨어의 종합적인 개발환경은 타겟에서의 동작환경과 호스트에서의 개발환경으로 크게 나뉘어 지며 실제로 적용할 수 있는 개발환경을 구축하려면 다양한 동작환경과 개발환경의 조합이 매우 다양하여 복잡하다. 타겟에서의 동작환경에는 타겟의 운영체제의 종류, 임베디드 프로세서를 포함한 하드웨어 플랫폼 등이 매우 다양하다. 임베디드 소프트웨어의 개발도구는 주로 PC/Windows 등 개발시스템에서 동작하지만 교차개발도구이므로 타겟의 동작환경과 밀접한 연관이 있다.

이러한 임베디드 소프트웨어의 개발환경의 다양성으로 처음에는 타겟의 운영체제와 개발도구가 고유의 프로토콜로 연동되어 분리할 수가 없었다. 예를 들어 VxWorks/Tornado는 실시간 운영체제인 VxWorks과

그와 같이 연동되는 개발도구인 Tornado가 분리될 수 없는 한 패키지로 제공되었다.

다음으로는 타겟의 운영체제와 개발도구가 분리하여 일정한 인터페이스만 구현하면 서로 연동이 가능한 구조가 되었다. 교차디버깅 전문 도구인 SingleStep, Code Warrior 등이 그러한 구조이다. 국내의 ETRI에서 주관하여 개발한 Esto도 그러한 분리형 구조를 갖는 대표적인 개발도구이다. 이러한 분리형 구조는 다양한 임베디드 소프트웨어의 환경에 쉽게 적용할 수 있는 장점이 있다.

최근에는 MicroSoft의 Visual Studio에 맞서 누구든지 기능을 쉽게 추가할수 있는 개방형 구조(Open Architecture)의 공개소스형 소프트웨어 개발도구 협의회를 구성하였다. 2002년부터 IBM, Borland, Fujitsu, Hitachi, Montavista, ETRI, HP 등이 참가하여 Linux상에서 Java를 기반으로 한 종합개발도구를 정의하고 필요에 따라 기능을 쉽게 추가할수 있는 Plug-in 구조를 정의하고 개발하였다. Eclipse는 Java를 이용하여 개발되고 Java언어를 기본으로 지원하나 특별히 언어에 의존성이 적도록 설계하였다. 현재 Eclipse기반으로 플러그인 도구인 CDT(C/C++

Development Tools)를 공개 프로젝트로 개발하여 많은 회사들이 이를 임베디드 SW개발도구로 확장을 하여 자사 상품으로 내놓고 있다[3]. TimeSys의 TimeStorm[5], Montavista의 DevRocket[4], WindRiver의 Workbench[6]가 그것이다. MicroSoft의 Platform Builder는 Windows환경에서의 대표적인 개발환경인 Visual Studio의 축소판으로 임베디드 SW개발도구이다. ETRI의 Esto는 Visual Esto와 Eclipse Esto의 두가지 버전이 존재하는데 Visual Esto는 MS의 Visual Studio와 유사한 사용법을 갖는 임베디드 SW개발도구로서 소형인 반면 온칩디버깅 및 실시간 디버깅 등 다양한 기능을 갖추고 성능도 우수한 도구이며, Eclipse Esto는 Elipse기반으로 Visual Esto의 기능을 수용하는 종합도구이다.

Eclipse는 plug-in에 의한 다양한 도구를 연동시키는 것이 쉬우니 Visual Studio보다 초보자에게 사용법이 친숙하지 못하고 Java를 포함한 범용 개발도구로 정의되어 임베디드SW 개발을 처음부터 고려한 것이 아니므로 간단한 기능도 여러번 클릭해야 하는 단점이 있다. 또한 Java VM을 기반으로 구현하여 아직은 도구의 성능이 낮고 소형 응용에는 그 크기가 부담이 되는 수가 있다.

표 1. 세계적인 임베디드 소프트웨어의 개발도구 종류

개발환경 기능	Esto (ETRI, COSMO)	TimeStorm (TimeSys)	DevRocket (Montavista)	Platform Builder(MicroSoft)	Workbench (WindRiver)
사용법	Eclipse Visual Studio	Eclipse	Eclipse	Visual Studio	Eclipse
지원 OS(호스트/타겟)	Linux, Windows/ Embedded Linux, Qplus, RTOS, OS-less Firmware	Linux, Windows, Solaris/Embedded Linux	Linux, Windows, Solaris/Embedded Linux	Windows/WinCE	Linux, Windows/VxWorks, Embedded Linux
지원 타겟 CPU	ARM, Xscale, PPC, MIPS, X86	ARM, PPC, X86	ARM, Xscale, PPC, MIPS, X86, SH	ARM, Xscale, PPC, MIPS, X86, SH	ARM, Xscale, PPC, MIPS, X86
컴파일러	gcc, javac	gcc, javac	gcc, javac	MS 컴파일러	gcc, javac
디버거	gdb 확장 (non-stop/real-time 디버깅 지원)	gdb	gdb	MS 디버거	gdb 기반 자사 디버거

개발환경 기능	Esto (ETRI, COSMO)	TimeStorm (TimeSys)	DevRocket (Montavista)	Platform Builder(MicroSoft)	Workbench (WindRiver)
HW 디버깅(JTAG)	제공(저가)	지원안함	지원(고가)	지원(고가)	지원(고가)
Target Builder	고급/편리(의존성 자동 점검)	지원	지원	지원	지원
저전력 기능	제공	지원안함	지원안함	지원안함	지원안함
디바이스 드라이버 개발 도구	제공	없음	없음	제공	없음
실시간성 분석기	제공	없음	없음	없음	없음
편재형 시뮬레이터	개발 중	없음	없음	없음	없음

#### IV. Eclipse 기반 ETRI 통합개발환경 (Eclipse Esto)

ETRI에서는 2000년도 Eclipse 초기 년도부터 가입하여 2004년도부터 공개된 Eclipse 플랫폼과 CDT를 이용하여 임베디드 S/W 개발환경을 본격적인 개발에 착수하였다. 그림 2는 Eclipse 기반 ETRI 통합개발환경 (Eclipse Esto)의 내부 구성도이다. Eclipse Esto는 Eclipse 플랫폼에 임베디드 시스템 특성에 맞게 정의한 원격 개발 IDE(Integrated Development Environment), 원격 debugger, 타겟 event monitor 등을 개발하여 임베디드 S/W 개발환경의 기초를 구축하고 디바이스 드라이버 개발도구, 최적화 및 분석도구,

하드웨어 기반 디버거, 실시간성 분석도구 등을 추가하여 임베디드 S/W 개발의 통합도구로서 면모를 갖추고 있다.

특히 최적화 및 분석도구는 소스 수준의 성능, 코드 크기 그리고 전력소모량까지 분석하도록 하여 임베디드 SW의 성능, 메모리 사용량 등의 최적화 뿐만 아니라 유비쿼터스 환경에서의 전력소모량을 최소화하는 도구로 유용하게 사용될 수도 있다. Device driver 개발도구는 Linux를 기반으로 다양한 device의 종류에 따라 driver의 기본 구조 소스를 자동으로 생성하고 시험을 보조함으로써 각종 device driver 개발을 용이하게 하고 있다.

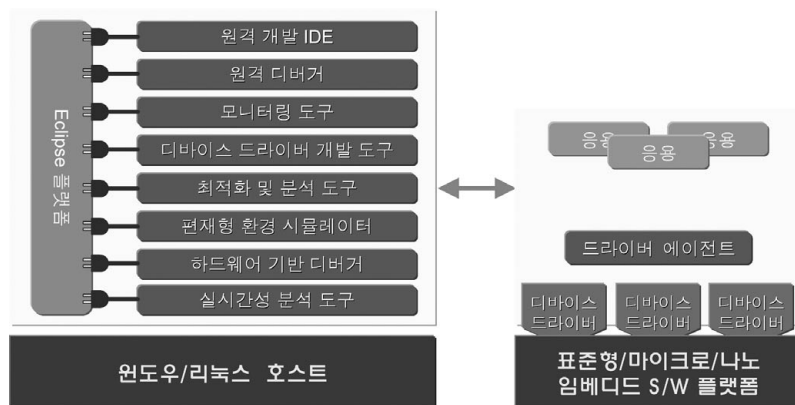


그림 2. Eclipse Esto의 구성도

또한 표준형, 마이크로형 및 나노형 Qplus 모두에 적용하도록 함으로서 동일한 환경으로 모든 개발환경에 사용되도록 설계되었다.

## V. Visual Esto의 상용화

Visual Esto는 ETRI와 (주)코스모의 공동 개발하여 (주)코스모에서 상용화하여 국내 교육기관 및 기업에 보급되고 있는 임베디드 S/W 통합개발도구이다. Visual Esto는 국내에서는 취약한 기반 소프트웨어 분야인 임베디드 시스템 소프트웨어가 세계적인 경쟁력을 갖게 된 유일한 제품으로서 다음과 같은 특징을 갖는다.

- Embedded 소프트웨어의 개발을 PC상에서 MS Visual Studio와 유사한 방법으로 용이하게 개발할 수 있다. 즉 초기에 임베디드 시스템의 특성만 정의해두면 개발자가 임베디드 시스템에 고유한 것을 몰라도 PC상에서 손쉽게 임베디드 S/W를 개발할 수 있도록 하였다.
- 임베디드 소프트웨어의 모든 단계, 즉 Linux 등 OS상의 응용 S/W개발 뿐만 아니라 OS내의 device driver 개발, on-chip debugging기술에 의한 firmware 개발 등 모든 단계에 세계 최초로 동일한 도구를 적용하도록 하였다.
- 비정지 정지점 설정, 디버깅 도중 실시간 타겟 제어 가능, 실시간 attach기능 등 다양한 실시간 디버깅 기능이 실현되어 있어 임베디드 시스템의 중요한 특징인 실시간성으로 실시간 동작 중의 오류도 디버깅할 수 있다.
- 실행환경(타겟/OS)와 개발환경(호스트/IDE)이 분리된 개방형 구조로서 동일한 도구로 Embedded Linux상에서의 대형 응용 뿐만 아니라 소형 real-time OS 응용 등 어떤 종류의 임베디드 시스템도 수용할 수 있어 ubiquitous 환경에 적합

하다.

- 타겟 모니터 및 트레이서, 임베디드 profiler, 고급 memory debugger 등 풍부한 시스템 종합시험 및 분석도구를 갖추고 있어 복잡한 응용시스템의 시험 기간을 효과적으로 단축한다.

## VI. 결론

임베디드 시스템의 특징과 관련하여 임베디드 S/W 개발도구의 중요성을 강조하고, 개발도구의 세계적 동향을 살펴보았다. 대부분의 국가 신성장동력 산업과 유비쿼터스 네트워킹의 공통 기반기술이라 할 수 있는 임베디드 소프트웨어의 효과적인 개발은 산업전반의 경쟁력에도 관련이 크다고 할 수 있다. 반면에 장기간의 기술 축적이 필요한 시스템소프트웨어 분야에는 극히 취약한 것이 국내 현실이다. 이러한 현실에서 임베디드 시스템 소프트웨어 분야에서 ETRI의 선도적 연구개발 성과와 업체의 수십년간의 통신시스템 등 임베디드 S/W 개발 경험이 효과적으로 협력하여 상용화에 성공한 Visual Esto가 국내 및 국외에서도 경쟁력을 갖기 시작하는 것은 산연의 좋은 본보기가 될 뿐만 아니라 u-korea를 주장하는 국가의 경쟁력을 실질적으로 한 단계 높이는 청신호가 되리라 기대한다.

## 참고문헌

- [1] 조주현, "임베디드 실시간 시스템의 개발환경," 한국정보처리학회 제9권 제1호 임베디드 시스템 특집, Jan. 2002.
- [2] Eclipse Foundation, <http://www.eclipse.org>

[3] ETRI, 2004 평가보고서, 2004.12.

[4] Jacob Lehrbaum, "MontaVista DevRocket: Building an Eclipse-based development environment for Embedded Linux," 2004.

[5] TimeSys Corp., <http://www.timesys.com>

[6] Wind River, Workbench User's Guide.

[7] COSMO Inc., Visual Esto User Guide. 