

# ETT<sup>esto</sup>: Eclipse 플러그인 기반 임베디드 통합 모니터링 환경<sup>+</sup>

윤남식\*, 박윤용\*, 배지혜\*, 임동선\*\*, 김재명\*\*

\*선문대학교 컴퓨터공학부

\*\*한국전자통신연구원 융합소프트웨어연구본부 조선융합기술연구팀

## ETT<sup>esto</sup>: Eclipse plug-in based Integrated Monitoring Environment for Embedded System

Nam-Sik Yun, Yoon-Young Park, Ji-Hey Bae

{windsong, yypark, angdoo98}@sunmoon.ac.kr

Dong-Sun Lim, Jae-Myoung Kim

{dslim, jaemkim}@etri.re.kr

### 요 약

임베디드 시장의 급속한 성장은 임베디드 소프트웨어의 수요와 산업 비중을 크게 증가시켰다. 임베디드 시스템의 복잡성은 점점 증가하는 반면, 시장에서 요구하는 개발 기간은 점점 짧아지고 있다. 이로 인해 효율적인 임베디드 소프트웨어 개발 기술 및 개발 플랫폼, 최적화를 위한 분석 도구에 대한 수요도 날로 증가하고 있다. 본 논문에서는 임베디드 시스템을 위한 소프트웨어 통합 개발 환경인 Esto 플랫폼 기반의 ETT<sup>esto</sup> 플러그인을 통한 타겟 시스템 분석 및 모니터링 환경을 제안한다.

### 1. 서론

임베디드 소프트웨어의 적용 분야는 휴대폰, 항공기, 자동차, 로봇, 각종 산업용 및 의료 기기, 첨단 무기 시스템에 이르기까지 다양한 분야에 걸쳐 확대되고 있다. 임베디드 산업의 급속한 성장에 따라 관련 운영체제 및 응용 소프트웨어, 마이크로프로세서, 개발 및 분석 도구에 대한 수요도 급격히 증가하고 있다. 이에 따라 정부도 임베디드 소프트웨어 기술을 신성장동력 분야의 하나로 선정하여 핵심기술 개발 및 산업기반 조성에 노력해왔다.

임베디드 소프트웨어는 크게 임베디드 운영체제,

미들웨어 기술, 사용자 인터페이스 기술, 개발 도구 등이 있으며, 하드웨어 시스템이 전체 시스템 성능을 좌우했던 과거에 비해 그 중요성이 점점 더 강조되고 있다.

임베디드 시스템의 자원적 제약으로 인한 교차 개발 환경의 필요성 및 타겟 시스템에 따른 독자적인 개발 환경의 특수성은 임베디드 소프트웨어 개발 및 시스템 모니터링을 어렵게 만든다. 더욱이 임베디드 시스템의 복잡성은 날로 증가하는 반면, 시장에서 요구하는 개발 기간은 점점 짧아지는 추세이다. 따라서 효율적인 임베디드 소프트웨어 개발을 지원하는 통합 개발 도구 및 최적화를 위한 분석 도구에 대한 수요도 날로 증가하고 있다.

본 논문에서는 한국전자통신연구원에서 개발한 임베디드 소프트웨어 통합개발환경인

+ "본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2008-C1090-0801-0020)

Esto(Embedded System Tool) 플랫폼을 기반으로 하는 eclipse 플러그인을 통한 임베디드 통합 모니터링 환경을 제안한다. 2장에서는 임베디드 소프트웨어 개발 환경에 대한 기존 연구들을 살펴보고, 3장에서는 임베디드 시스템 모니터링을 위한 ETT<sup>plus</sup> 시스템을 기술하고, 4장에서는 eclipse 플러그인을 통한 ETT<sup>esto</sup>를 제안한다. 마지막으로 5장에서는 본 연구에 대한 결론 및 문제점을 기술한다.

## 2. 관련 연구

### 2.1. 임베디드 소프트웨어 개발 환경

임베디드 시스템은 특정한 작업 수행을 위해 설계된 하드웨어와 소프트웨어로 구성된 솔루션으로, 범용 목적의 컴퓨터 시스템에 비해 제한된 성능을 가지고 있다. 따라서 소프트웨어 개발을 위한 컴파일러 및 라이브러리 등의 개발 환경 구축이 어렵고, 메모리 및 컴퓨팅 파워의 부족 등 임베디드 시스템에서의 직접적인 소프트웨어 개발이 불가능하다. 이로 인해 임베디드 시스템에서 수행될 소프트웨어를 개발하기 위해서 호스트 컴퓨터를 사용하고 있다.

임베디드 소프트웨어 개발은 타겟 임베디드 시스템에 비해 풍부한 자원과 강력한 컴퓨팅 파워를 가지는 호스트 컴퓨터에서 이루어지며, 이러한 개발 환경을 교차개발환경(Cross Development Environment)이라고 한다. 호스트 컴퓨터의 교차개발환경은 타겟 임베디드 시스템용 실행파일을 생성하는 크로스 컴파일러, 라이브러리 및 기타 유틸리티 등으로 구성된다. 이러한 호스트 컴퓨터의 교차개발환경을 통해 타겟 보드용 실행 파일을 생성하고, 개발된 소프트웨어는 타겟 보드로 전송하여 수행되게 된다.

그러나 명령어 라인 기반의 교차개발환경을 통한 소프트웨어 개발 방식은 소프트웨어 개발에 있어 불편함을 야기할 뿐만 아니라 개발 시간을 증가시키는 요인으로 작용한다. 또한 타겟 보드에 따른 독자적인 크로스 컴파일러를 요구하며, 이로 인해 임베디드 시스템의 복잡성이 증가하고 개발 환경의 유연성을 저해하게 된다. 따라서 일관되고 사용하기 쉬운 임베디드 시스템을 위한 개발 환경에

대한 중요성이 점점 커지고 있다.

통합개발환경(Integrated Development Environment)은 일관된 사용자 환경, 쉬운 인터페이스, 체계적인 프로젝트 관리 기능 등 개발상의 편의를 제공하며, 이를 통해 소프트웨어 개발의 생산성과 코드 완성도를 향상시킬 수 있다[1],[2].

임베디드 소프트웨어 개발을 위한 통합개발환경은 개발 타겟 보드의 환경설정 및 교차개발환경 구축, 코딩 및 크로스컴파일, 타겟 보드로의 실행 파일 전송 및 원격 실행, 원격 디버깅, 원격 모니터링 등의 강력한 기능을 제공하며, 이를 통해 임베디드 소프트웨어 개발의 편의성 및 효율성, 개발 시간 단축 효과를 기대할 수 있다.

임베디드 시스템 개발을 위한 교차개발환경을 지원하는 도구로 Wind River Workbench, QNX's Momentics, Timesys's TimeStorm, Montavista's DevRocket 등이 있으며, 국내에서는 한국전자통신연구원에서 개발한 Esto, Visual Esto 등이 시장에 나와 있다.

### 2.2. 임베디드 시스템 모니터링 환경

임베디드 소프트웨어는 한정된 자원을 가지는 시스템에서 특정 작업을 수행하기 위한 응용프로그램이다. 따라서 소프트웨어 개발 단계에서부터 최적화를 위한 고려가 필요하며, 이를 위한 디버깅 및 시스템 모니터링 환경의 중요성이 날로 커지고 있다.

기존 리눅스 시스템에서는 LTT(Linux Trace Toolkit), LTTng(LTT Next Generation) 등의 커널 추적 기반의 모니터링이 가능하였다. 이러한 추적 도구는 커널 패치를 통해 시스템의 주요 이벤트들을 추적하여 샘플링 된 추적 데이터를 분석, 시각화 도구를 통해 제공한다[3],[4],[5].

그러나 이러한 기존의 커널 추적 도구는 커널 버전과 패치에 따른 복잡성 및 호환성 문제를 가진다. 또한 임베디드 시스템이라는 특수한 환경 요소를 반영하지 못한다. 이러한 문제 해결을 위해 임베디드 시스템 환경에서의 시스템 모니터링을 위한 ETT<sup>plus</sup> 시스템을 제안하였으며, 이를 활용한 임베디드 시스템 성능 측정을 검증한 바 있다[6].

임베디드 소프트웨어 개발에 있어서 통합개발환경 및 개발 도구의 비중이 커지면서 임베디드 소

소프트웨어 디버깅 및 시스템 모니터링 도구도 통합 개발환경에 포함된 형태로 제공하는 추세이다. 한국전자통신연구원에서 개발한 Esto 또한 기본적인 소프트웨어 개발 이외에 임베디드 시스템 모니터링, 전력 분석 및 최적화 도구를 제공하고 있다.

본 논문에서는 eclipse 플랫폼 기반의 임베디드 소프트웨어 통합개발환경을 제공하는 Esto에 플러그인 형태로 ETT<sup>plus</sup> 시스템을 접합하여 소프트웨어 개발 및 모니터링이 가능한 통합개발환경을 제안한다.

### 3. ETT<sup>plus</sup>

ETT<sup>plus</sup>(Embedded kernel Trace Toolkit) 시스템은 임베디드 시스템 모니터링 및 성능 분석을 위해 고안된 임베디드 리눅스 커널 기반 추적 도구이다. 그림 1은 ETT<sup>plus</sup> 시스템의 전체 구조를 나타낸다.

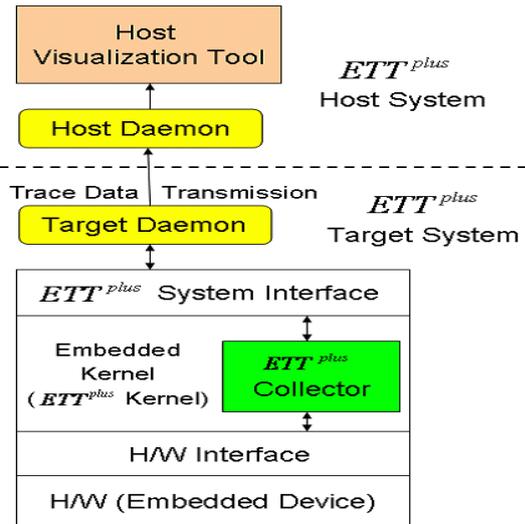


그림 1. ETT<sup>plus</sup> 시스템 전체 구조

ETT<sup>plus</sup> 시스템은 임베디드 시스템의 특수한 환경을 고려하여 타겟과 호스트 시스템으로 구성된다. 타겟과 호스트 간의 네트워크 통신을 통해 추적 데이터 및 제어신호를 전송하고, 타겟 커널 내부에서 생성되는 샘플링 데이터의 저장은 메모리를 사용하여 시스템 부하를 최소화 하였다. 타겟 보드에서 구동되는 임베디드 리눅스 커널은 데이터 추적을 위해 패치 된 커널이 사용되며, 호스트의 시각화 도구는 GTK+ 기반 GUI를 제공한다.

커널 내부에서 발생하는 각종 이벤트는 샘플링

과정을 거쳐 타겟 데몬(Target Daemon)에 의해 수집, 네트워크를 통해 호스트 시스템에 전송되며, 호스트 시스템의 시각화 도구를 통해 분석된 정보가 제공되는 구조를 가진다. ETT<sup>plus</sup> 시스템의 자세한 설명은 과거 연구[6]에 자세히 기술되어 있다.

ETT<sup>plus</sup> 시스템은 타겟 시스템의 프로세스 정보, 인터럽트 및 시스템콜 추적을 통해 임베디드 시스템에서 구동되는 소프트웨어의 정보를 수집, 이에 대한 분석 정보를 제공한다. 또한 타겟 시스템 관리를 위한 추가적인 원격 제어 기능을 가진다.

### 4. ETT<sup>esto</sup>

한국전자통신연구원에서 개발한 Esto는 임베디드 리눅스 기반 운영체제인 Qplus 플랫폼에서 수행되는 펌웨어, 디바이스 드라이버, 응용 등의 임베디드 소프트웨어 개발을 지원하는 eclipse 플랫폼 기반의 통합개발환경을 말한다.

Esto는 기본적인 임베디드 소프트웨어 개발 기능 및 크로스컴파일, 원격 실행 및 원격 디버깅 등의 강력한 기능을 제공한다. 또한 루프 수준 분석을 통한 전력 측정 및 코드 최적화, 타겟 시스템의 메모리, 프로세스 자원 정보 등의 시스템 모니터링 기능을 제공한다. 또한 eclipse 기반 플랫폼을 채택하여 추가 플러그인(plugin)을 통한 기능 확장의 유연성을 가진다.

Esto와 ETT<sup>plus</sup>에서 제공하는 타겟 임베디드 추적 기능은 그림 2와 같다. Esto와 ETT<sup>plus</sup> 모두 타겟 시스템의 관리를 위한 브라우징 및 원격 제어 기능을 제공한다. Esto는 타겟 시스템의 메모리 사용에 대한 풍부한 정보를 제공하며 IPC 및 전력 소모량을 측정할 수 있다. ETT<sup>plus</sup>는 타겟 시스템의 커널 내 시스템콜 레벨의 추적이 가능하며, 이에 기반을 둔 프로세스 분석 기능을 제공한다.

	Esto	ETT <sup>plus</sup>
타겟브라우징 및 원격제어	지원	지원
CPU 정보	지원	지원
메모리 정보	메모리맵, 메모리사용 정보	메모리맵
프로세스정보	지원	지원
시스템콜	-	지원
인터럽트	지원	지원
IPC	지원	-
전력량측정	지원	-

그림 2. Esto와 ETT<sup>plus</sup>의 추적 기능 비교

본 논문에서 제시하는 ETT<sup>esto</sup>는 ETT<sup>plus</sup> 기능을 가지는 eclipse 플러그인을 뜻한다. 이를 통해 기존 Esto에서 제공하지 못했던 커널 내부의 시스템클레벨의 프로세스 추적 및 분석이 가능하며 시스템 성능 분석 및 임베디드 소프트웨어 최적화를 위한 분석 데이터 제공이 가능하다. 그림 3은 이러한 ETT<sup>esto</sup>의 컨셉을 나타낸다.

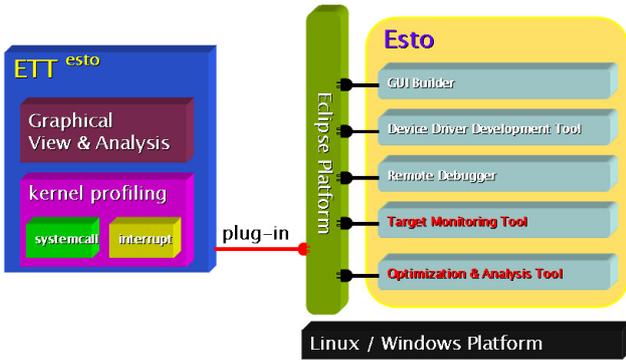


그림 3. eclipse 기반 ETT<sup>esto</sup> 플러그인

그림 4는 eclipse 플랫폼에 플러그인 형태로 구현된 ETT<sup>esto</sup>이다. 기존의 ETT<sup>plus</sup>의 시각화 분석 도구를 Java 언어로 재구성하여 타겟 시스템의 추적 결과를 분석해 제공한다. ETT<sup>esto</sup>는 eclipse 플랫폼을 사용한 Esto에 플러그인으로 제공되어 임베디드 소프트웨어 개발에 필요한 시스템 모니터링 및 성능 분석, 이에 기반을 둔 최적화 도구를 지원하는 통합개발환경 구축이 가능하다.

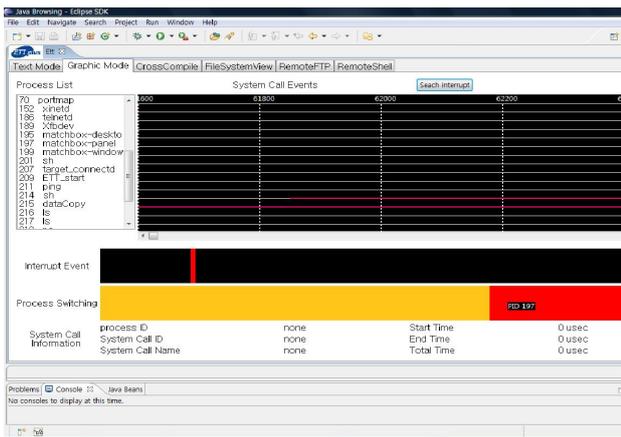


그림 4. eclipse 플러그인 형태의 ETT<sup>esto</sup>

## 5. 결론

본 논문에서는 임베디드 소프트웨어 개발환경 및 모니터링 환경에 대해 살펴보았다. 또한 임베디드

드 시스템 모니터링을 위한 기존 연구인 ETT<sup>plus</sup> 시스템에 대해 소개하였으며, 이를 이용한 ETT<sup>esto</sup>를 제안하였다.

ETT<sup>esto</sup>를 통해 기존 Esto에 eclipse 플러그인 형태로 추가 추적 기능을 제공하여 시스템 모니터링 및 성능 평가, 최적화 작업을 위한 통합모니터링환경을 제안하였으며, 이를 통해 더욱 효과적인 임베디드 소프트웨어 개발, 임베디드 시스템 분석 및 최적화 작업을 가능케 하였다.

그러나 ETT<sup>esto</sup>는 기존의 ETT<sup>plus</sup> 시스템과 마찬가지로 미리 패치된 타겟 시스템의 커널을 기반으로 동작하며, 결과적으로 시스템의 유연성과 호환성을 저하시키는 단점을 가진다. 또한 타겟 시스템의 실시간 모니터링을 지원하지 못하는 문제점을 가진다. 이러한 문제는 차후 연구를 통해 해결해야 할 과제이다.

## [참고문헌]

- [1] P.Varhol, "Integrated Software Tools Improve Productivity and Code Quality", Electronic Design, Oct. 1999, pp.62~70.
- [2] S.V. Tyle, "Engineering software Tools Meet Demands", Electronic Design, June 1994. pp.71~80.
- [3] Opersys Homepage, <http://www.opersys.com/LTT>
- [4] LTTng & LTTV Homepage, <http://ltt.polymtl.ca>
- [5] Mathieu Desnoyers, Michel R. Dagenais, "The LTTng tracer : A low impact performance and behavior monitor for GNU/Linux", Linux Symposium, Ottawa, Canada, Jul, 2006.
- [6] 배지혜, 윤남식, 박운용, "임베디드 커널 추적 도구를 이용한 임베디드 시스템 성능 측정 기법", 정보과학회논문지:컴퓨팅의 실제 및 레터, 제13권 제7호, 2007년 12월, pp.462~475.