

임베디드 SW의 시험 환경 구축

김금옥*, 양해술**, 김철원*
*호남대학교 정보통신대학
**호서대학교 벤처전문대학원
e-mail : tyhjc@naver.com

Embedded SW's examination environment construction

Jin, Jin-yu*, Hae-Sool, Yang**, Chul-Won, Kim*
*College of Information and Communication, Honam University
**Graduate School of Venture, Hoseo University

요 약

최근 임베디드 소프트웨어가 다양한 분야에서의 개발이 증대되면서 요구사항도 변화되고 있다. 즉, 임베디드 시스템이 단순하고 독립적인 소형시스템에 정착되어 운영되었던 과거와는 달리 최근에는 초정밀의 동작제어와 함께 복잡하고 다양한 플랫폼 환경을 요구하거나 다수의 소프트웨어를 통합하고 제어하는 시스템 중속적인 요구사항들이 많이 발생하고 있다. 본 연구에서는 임베디드 소프트웨어 분야의 기반기술 및 동향을 조사하고 표준시험환경을 구축해 보았다.

1. 서론

임베디드 소프트웨어는 초기에는 간단한 제어 프로그램만으로 산업용 기기를 제어하는데 그쳤으나, 최근에는 멀티미디어 처리와 같은 점차 복잡한 기능을 요구하면서 임베디드 운영체제를 사용하기 하였고 멀티태스킹, 네트워크 기능을 제공하는 임베디드 운영체제를 탑재하여 인터넷을 통해 제품에 대한 감시가 가능해졌으며, 나아가 임베디드 시스템에 탑재된 유무선 통신망을 통해 수시로 업데이트할 수 있는 임베디드 소프트웨어 플랫폼이 출현하기에 이르렀다.

최근 임베디드 소프트웨어는 유.무선 통신과 디지털 정보기기를 기반으로 하여 홈 네트워킹과 인터넷 정보기전에 의한 스마트 홈(Smart Home) 시대가 바야흐로 도래되었으며, 이는 점차 발전하여 유무선 인터넷과 웨어러블 컴퓨터에 의한 스마트 시티즌(Smart Citizen) 시대를 거쳐 도시 생활 환경 속의 유비쿼터스 컴퓨팅에 의한 SmartTown (Smart Town) 등의 영역으로 그 활용 분야가 급격히 확장되어 「Embedded, Everywhere」 시대로 발전할 전망이다. 따라서, 임베디드 소프트웨어 산업에 대한 중요성의 증대와 경쟁 우위를 점하기 위한 품질 확보의 필요성이 요구됨에 따라 임베디드 소프트웨어에 대한 품질평가 시험 환경 구축에 관한 필요성이

고조되고 있다. 본 연구에서는 임베디드 소프트웨어 분야의 기반기술 및 동향을 조사하고 표준시험환경을 구축해 보았다.

2. 관련 연구

2.1 임베디드 소프트웨어 적용분야 및 특징

임베디드 소프트웨어는 다음과 같은 특성을 가진다. 첫째로, 임베디드 소프트웨어가 실행되는 시스템의 용도에 따라 연성 혹은 경성 실시간 처리를 지원하여야 한다. 예를 들어 무인항공기에 사용되는 비행제어시스템, 항법시스템 등에 내장되는 임베디드 소프트웨어는 무인 항공기 제어에 허용되는 시간 내에 작업을 처리하지 않으면 큰 손실을 초래할 수 있다.

둘째로, 소프트웨어 오동작 및 작동 중지가 허용되지 않는 임베디드 시스템에서는 고도의 신뢰성이 요구된다. 원자력 발전, 항공기 제어, 미사일 등과 같이 mission-critical한 임베디드 시스템에서는 소프트웨어 오동작 또는 불시의 작동 중지 등의 심각한 결과를 초래한다.

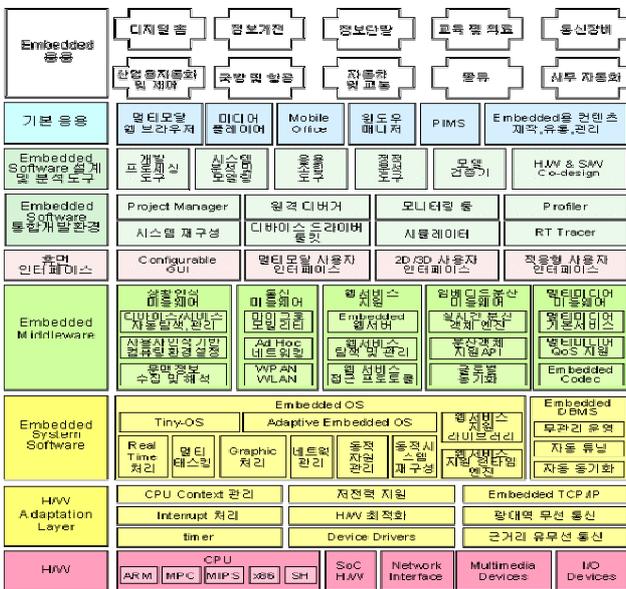
셋째로, 임베디드 시스템은 크기, 가격 및 발열 등을 이유로 제한된 하드웨어 자원으로 구성되므로 임베디드 소프트웨어는 경량화, 저전력 지원, 자원의 효율적 관리 등의 하드웨어에 최적화되는 기술을 지

원하여야 한다.

넷째로, 범용 데스크탑 또는 서버에서 실행되는 패키지 소프트웨어와 달리 특정 시스템의 실행을 목적으로 개발되는 소프트웨어라는 점에서 차이가 있다.

다섯째로, 최근 임베디드 시스템의 기능이 다양해지면서 임베디드 시스템들이 stand-alone 시스템뿐만 아니라 유·무선 네트워크를 통해 연결될 수 있어야 하며, 디지털 TV, PDA, 스마트 폰 등의 임베디드 시스템에서 텍스트 위주에서 멀티미디어 정보들을 처리하는 기술이 필요로 함에 따라 네트워크 및 멀티미디어 처리 기능을 지원하여야 한다.

임베디드 소프트웨어의 적용분야는 아래 (그림 1)과 같다.



[그림 1] 임베디드 소프트웨어 적용분야

2.2 임베디드 SW 표준화 동향

2.2.1 임베디드 시스템 SW 및 하드웨어 분야

임베디드 하드웨어 분야는 임베디드 시스템용 마이크로프로세서가 고성능화 되고 있으며, 저가의 휴대형 정보기기를 위한 고성능, 저전력 임베디드 시스템의 요구가 증가하면서 Configurable IP 및 Configurable SoC 플랫폼 기술 개발로 발전하고 있다.

VxWorks, pSOS, QNX와 같은 전용 실시간 OS는 지원 기능을 다양화하여 새로운 요구에 부응하여야 하는 과제에 직면하여 있으며 WindRiver사는 BSD의 소유권을 사들여 VxWorks의 영역 확장을

피하고 있다. 유럽에서는 Psion, 에릭슨, 노키아 등의 연합체인 심비안이 소형 휴대 전화기용 임베디드인 EPOC을 개발하였으며 객체지향설계, Pen 기반의 GUI 및 풍부한 응용 지원을 특징으로 한다. MS사는 스마트 폰과 같은 모바일 제품용 플랫폼으로 스틱어(Stinger)를 정의하고 기본 OS인 WinCE 뿐만 아니라 모바일 브라우저, GSM, 음성 통신 등과 같은 기능을 OS에 포함하여 제공하고 있으며 국내에서는 CDMA, IMT 2000 등을 추가 지원하여야 할 것이다. MS사는 멀티미디어 솔루션을 위해 멀티미디어 라이브러리로서 DirectX를 제공하여 멀티미디어 응용의 기반 소프트웨어로 WinCE가 채택되는 중요한 요소로 작용하고 있다.

2.2.2 임베디드 미들웨어 분야

임베디드 미들웨어의 위치기반 상황인식 서비스는 모바일 컴퓨팅 환경에서 수시로 움직이는 객체인 단말과 사용자의 위치를 인식하여 다양한 위치 인식 응용에 활용할 수 있도록 하는 것으로, MS의 UPnP(Universal Plug and Play), Sun의 Jini, Oracle의 9iASWE(Oracle9i Advanced Search Wireless Enabling), CMU의 Coda, UCL의 XMIDDLE (Information Sharing Middleware for Mobile Environment) 등이 있다. 이동성 지원 기술은 고속의 WLAN 개발과 WPAN(Wireless Personal Area Network)의 활용이 높아짐에 따라 빠른 로밍과 마이크로 모빌리티의 연구가 활발하게 수행되고 있다.

시스템 및 네트워크 환경에 적응하여 서버로부터 클라이언트로 다운로드되는 멀티미디어 스트리밍 기술은 MS의 DirectX 8.1과 WMF(Windows Metafile, SUN의 JMF(Java Media Framework), Real Networks의 Real Player 등에서 개발 중에 있다.

분산 실시간 미들웨어 기술은 SUN과 IBM이 공동으로 연구하고 있는 Real-time JAVA를 비롯하여 HP의 J-Consortium, OMG(Object Management Group)의 RealTime-CORBA, Washington Univ.의 TAO1)등에서 연구가 활발하게 이루어지고 있다.

3. 임베디드 소프트웨어의 시험환경 구축

이 절에서는 임베디드 소프트웨어를 시험하기 위

해 ISO/IEC 9126과 ISO/IEC 12119에 근거하여 임베디드 소프트웨어의 평가항목을 도출하고 임베디드 소프트웨어 시험환경과 시험절차를 구축하였다. 그리고 본 연구에서 개발한 시험체계를 도입한 평가 사례에 대하여 기술하였다.

3.1 임베디드 소프트웨어 시험 절차

임베디드 소프트웨어 시험 평가를 위해서는 먼저 제품에 대한 정확한 이해가 필요하며 제품을 정확히 분석하여 테스트케이스를 구성하여야 한다. [표 2]과 같이 임베디드 소프트웨어의 품질평가 체계와 매트릭은 객관적인 시험을 통해 시험 대상 제품이 소프트웨어 품질 요구사항을 만족하는지 확인하고, 소프트웨어 품질을 개선함으로써 제품의 신뢰도 향상 및 경쟁력을 향상 시킬 수 있다. 본 연구에서 제안한 임베디드 소프트웨어 시험 평가를 위해서는 다음과 같은 방법을 취할 수 있다.

[표 2] 임베디드 소프트웨어 시험절차

구분	세부 시험절차
1	시험 계획 수립
2	시험 환경 구축
3	시험 항목 도출
4	테스트 시나리오(TS)와 테스트 케이스(TC) 작성
5	시험 수행
6	결함 수정
7	회귀시험 수행
8	공장 심사 수행
9	시험결과서 및 평가결과보고서 작성

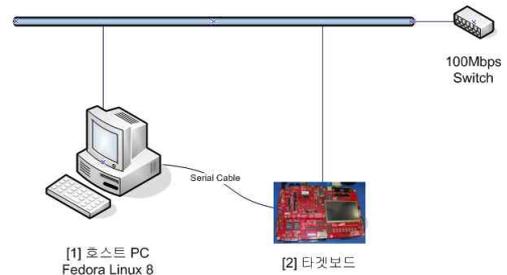
먼저 시험을 하고자 하는 제품의 시험 계획을 수립하고 다음은 시험 환경을 구축하여 시험 환경을 구성할 수 있도록 하고 제안한 매트릭에 대하여 시험 항목을 도출하며 테스트 시나리오와 케이스를 작성하고 시험을 수행한다.

시험 도중 발생하는 결함에 대해서는 결함리포트를 작성하고 작성된 결함리포트는 결함리스트를 중심으로 하여 제품의 패치를 통해서 결함을 수정하고 다시 제품에 대한 회귀시험을 수행하여 시험결과를 작성한다.

3.2 시험 환경

시험환경에서 시험할 제품은 리눅스 기반 Qplus 운영체제에서 운용되는 C/C++ 응용어플리케이션을

개발하기 위한 이클립스 CDT(C/C++ Development Toolkit) 기반의 임베디드 응용어플리케이션 개발 프로그램으로 주요기능으로는 프로젝트 관리, 실행환경 설정기능, Debug, Build 및 Run 기능, 타겟 시스템 자원 모니터링 기능 등이 있으며 이 절에서는 임베디드 소프트웨어를 시험하기 위해 (그림 2)와 같이 시험환경을 구축하였다.



[그림 2] 시험환경 구축

(그림 2)와 같이 호스트 PC에 설치한 프로그램으로는 시험 대상 제품(개발도구)인 Eclipse CDT, ToolChain(gcc for ARM)을 설치하였고, [2]번 타겟 보드에 설치한 프로그램 또한 시험 대상 제품 (Agent/gdbserver)인 Qplus 운영체제를 설치하였다. 네트워크는 10/100Mbps 스위치와 시리얼 Cable을 사용하였다.

[표 2] 세부사양

No	Role	OS	CPU	Mem	HDD	Pre-Requriste
1	호스트 PC	Fedora Linux 8	Intel Core2Duo 2.13GHz *2	2GB	500GB	- eclipse CDT - ToolChain (gcc for ARM)
2	타겟 보드	Qplus	S3C6400	SRAM 512KB	NAND Flash 64MB	- SMDK6400

성능을 측정 하기 위한 성능측정도구는 시험대상 제품의 자원 사용률을 측정하기위해 TeamQuest Manager를 [1]번 호스트 PC에 설치하여 임베디드 소프트웨어의 시험환경을 구축하였다.

4. 시험 사례

4.1 시험 항목

ISO/IEC 9126(소프트웨어 품질특성과 매트릭에 관한 국제표준)과 ISO/IEC 12119(소프트웨어 패키지의 품질요구사항 및 시험에 관한 국제표준)에 근거하여 임베디드 소프트웨어의 시험항목을 도출하였다.

[표 1] 시험항목

주특성	부특성	설 명
기능성	적합성	사용자의 목적하는 바에 따라 적합한 기능을 제공하는 능력 및 프로그램의 적합한 구현 능력
	정확성	문서의 정확한 정보제공 능력과 프로그램의 정확한 구현 능력

신뢰성	성숙성	이전 버전에 대한 문제점을 해결할 수 있는 능력 및 결함으로 인한 기능장애를 피할 수 있는 능력
	결함허용성	결함이나 인터페이스 문제 발생시에도 지정된 수준의 성능을 유지하는 능력

효율성	시간효율성	주어진 조건에서 기능을 수행할 때 반응시간, 처리시간, 처리율을 제공하는 능력
	자원효율성	주어진 조건에서 기능을 수행할 때 I/O 자원, 메모리를 사용 하는 능력 및 데이터를 전송할 수 있는 능력
사용성	이해가능성	사용자가 SW를 활용하기 위한 방법이나 조건, 적절성 등을 파악할 수 있게 하는 능력
	학습성	사용자가 SW의 용법을 배울 수 있게 하는 능력과 도움말을 사용할 수 있는 능력

유지보수성	분석성	발생하는 에러의 증상 및 장애원인을 진단하고 변경될 부분을 식별 할 수 있게 하는 능력
	변경성	환경 설정이 변경 될 수 있게 하는 능력
유지보수성	안정성	환경 설정 변경에 따른 예상 밖의 결과에 대한 정보를 제공 하는 능력
	시험가능성	SW가 스스로 시험할 수 있는 능력 (내장형 시험기능)

이식성	적응성	SW가 요구하는 시스템 환경에 적응하는데 필요한 최소한의 조치만으로 이식될 수 있는 능력 및 데이터 구조를 바꿀 수 있는 능력
	설치가능성	지정된 환경에 설치, 제거될 수 있는 능력

일반적 요구사항	식별및표시	
	안전성	

4.2 시험결과

[표 3] 시험결과

순번	시험환경 OS	결함 요약	결함 정도	품질 특성	결함 설명
1	-	제품 지원 정보 미제공	M	일반적 요구사항	[매뉴얼] 제품운영을 위한 지원정보(예: 고객지원센터 연락처)가 제공되지 않음
2	-	제품 설치 정보 미제공	M	일반적 요구사항	[매뉴얼] Agent/gdbserver 관련 설치 및 설정에 관한 정보가 제공되지 않음
3	-	매뉴얼 철자 오류	L	사용성	[매뉴얼] 매뉴얼 39p 의 "2. 빌드 옵션 설정" 첫째줄에 철자 오류(예) gcc를 이용해 컴파일 때(x)
4	시험환경은 OS	진행상태 표시 오류	M	사용성	[esto>progress] 파일 'RUN'실행시 progress바 상태가 항상 "61%"로 표시됨
5	시험환경은 OS	binaries 폴더 미생성	H	기능성	[Esto>File>Import] <시나리오> 1.[File>NewProject>EstoProject]의 C++ project 선택 2.프로젝트 명 입력 후 Tochain의 Esto Arm Gcc 선택 하여 새로운 프로젝트 생성 3.[Import>FileSyste

					m>estoworkspace]의 "face" 폴더 선택 =>Build시 Binaries 폴더가 생성되지 않음
6

4. 결론

임베디드 소프트웨어 산업에 대한 중요성의 증대와 경쟁 우위를 점하기 위한 품질 확보의 필요성이 요구됨에 따라 임베디드 소프트웨어에 대한 품질평가 시험 환경 구축에 관한 연구의 중요성을 대두 되고 있다.

본 연구에서는 임베디드 소프트웨어의 개발환경 및 기술동향을 조사하여 임베디드 소프트웨어 시험 환경을 구축을 통해 임베디드 소프트웨어의 품질관리 체계를 확립하였다. 향후 실질적인 임베디드 소프트웨어의 시험환경 구축 활용을 통해 임베디드 소프트웨어 품질관리 표준화를 촉진함으로써 높은 부가가치를 창출하고 국제적으로 경쟁력을 갖춘 제품의 시험 및 개발을 지원할 수 있을 것으로 기대한다.

참고문헌

- [1] Embedded, Everywhere a Research Agenda for Networked Systems of Embedded Computers, National Research Council, National Academy Press, 2001.
- [2] Jean J. Labrosse, "Embedded Systems Building Blocks; Complete and Ready-to-Use modules in C", CMP books, 2000.
- [3] OPEN-JTAG 開發小組, " ARM JTAG 調試原理", pp1~9, 2004
- [4] "邊界掃描器件BSDL描述在測試中的應用", 半導體技術, pp1~4, 2006
- [5] "국내 개발체계 기반의 임베디드 소프트웨어 발전방향", ETRI, 소프트웨어 산업 부문별 시장/기술 전망 세미나, 2004.
- [6] 김백규, 임희연, "스마트 ARM 마이크로컨트롤러 AT91SAM7S ② 개발환경 구축 ", 임베디드 월드, pp 80~86, 2006
- [7] "임베디드 소프트웨어," LG주간경제, 2004.11.
- [8] "임베디드 소프트웨어 산업 실태조사"(KESIC), 2007.
- [9] 김창환, "임베디드 소프트웨어 추진동향" 정보통신연구진흥원, 주간기술동향 통권 1351호, 2008. 6. 18