

# Platform 기반의 임베디드 시스템 가상 프로토타이핑 》

이영란\*, 이정배\*, 이대선\*, 정영진\*  
\*선문대학교 컴퓨터정보학부

{yrlee, jblee}@sunmoon.ac.kr, sbkim@anybill.com

## A Virtual Prototyping of Embedded System Based on Platform

Young-Ran Lee\*, Jeoung-Bae Lee\*, Kang-Woo Han\*, Dae-Sun Lee\*,  
Young-Jin Jeoung\*

\*Sunmoon University, Division of Information & Computer Science  
{yrlee, jblee}@sunmoon.ac.kr

현재 디지털 기술의 핵심인 임베디드 시스템은 2010년 국민소득 2만불 달성이라는 국가 발전 목표의 실현을 위한 핵심 산업으로 2003년 8월 10개 신성장동력 산업의 핵심 육성 분야로 선정되었다. 하지만, 특수한 목적을 위한 다양한 임베디드 시스템은 개발 및 유지 보수 비용과 개발 기간이 장기화 됨으로써 시장경쟁력 약화의 원인이 된다.

본 논문에서는 임베디드 시스템의 신뢰성 있는 개발을 위한 방법으로 가상프로토타이핑을 제안한다. 소프트웨어와 하드웨어로 각각 구현될 도메인을 선정하여 플랫폼 라이브러리에 등록되고 통합된 시뮬레이션을 통한 검증은 수행함으로써 제조 공정이 다른 여러 기능의 블록을 집적하는 기술을 가상플랫폼을 기반으로 가상프로토타이핑한다.

### 1. 서론

컴퓨터 기술의 발전에 따라 임베디드시스템 소프트웨어는 범용 마이크로프로세서 및 IP(Intellectual Property)산업의 발전으로 임베디드 시스템의 요구에 따라 특수한 기능을 수행하는 설계로 산업의 전반으로 적용이 확대되고 있다. 그러나 점차 복잡도가 증가하는 임베디드 시스템 설계에서 제품의 기능 검증까지 소모되는 시간과 비용의 부담은 시스템의 초기 개발 비용을 증가시킴으로써 제품의 경쟁력을 저하시킨다. 따라서 시스템 개발시 동일 목표로 구현되는 시스템의 하드웨어와 소프트웨어의 역할 및 동시 설계와 하드웨어와 소프트웨어 설계를 위한 도메인, 검증과 디버깅과정이 개발방법론에 포함되어야 한다. 본 논문에서는 임베디드 시스템에 대한 이해와 하드웨어의 특성에 민감한 임베디드 시스템의 개발을 위하여 가상 프로토타이핑 방법을 제안한다. 임베디드 시스템 개발시 실제 시스템을 개발과정에 실제

하드웨어 및 마이크로프로세서에 최적화된 시제품을 개발하기는 현실적으로 어려움이 따르며, 오류 및 수정 기간이 장시간 소요됨으로써 제품 개발비용의 80%를 차지하는 초기개발 비용의 부담을 증가시켜 제품의 경쟁력 약화의 원인이 된다.

본 논문에서는 Virtual Platform기반의 임베디드 시스템 개발 도구인 Virtio를 이용한 UART 가상프로토타이핑을 구현하고 동작 테스트를 실행한다.

### 2. 가상프로토타이핑(Virtual Prototyping)

프로토타입(prototype)이란 제품 개발 과정 중 제품 출시 전에 제작한 축소형 또는 실물크기의 모델로, 제품의 외형 및 기능 시험을 위한 테스트용 모델을 제작하는 것으로 제품 생산과정의 일부로 볼 수 있다. CAD를 이용한 외형의 프로토타이핑은 제품개발과정으로 널리 이용되고 있으며, 공학적 관점에 기능에만 중점적으로 프로토타이핑 또한 제품 개발 현장에서 활용되어 왔다. 하지만 제품의 시각적, 공학적 관점을 병행하여 하드웨어와 소프트웨어의 co-design과 co-simulation을 수행하는 프로토타이

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구 결과로 수행되었음

평은 초기 단계라 할 수 있다. 프로토타이핑을 컴퓨터상의 데이터로 구현하여 기능 테스트와 검증 및 설계상 오류 수정 및 동작 재설정을 위한 유연함과 편리함을 제공하는 것을 가상 프로토타이핑이라고 한다.

가상프로토타이핑은 제품의 초기 비용과 개발시간과 소요인력을 최소화함으로써 Time-To-Market을 단축함으로써 경쟁력을 향상하는데 목적이 있다.

### 2.1 임베디드 시스템 가상프로토타이핑 환경

임베디드 시스템은 특수한 기능을 위하여 내장되는 시스템으로 용도가 제한적이며 크기, 메모리, CPU, 네트워크 대역폭, 저전력, 실시간 등 여러 제약사항을 고려한 선택이 되거나 개발이 진행되어야 한다. 이러한 제약사항을 고려한 자원 모델 기반의 방법 및 하드웨어 소프트웨어의 동시 개발방법에 관한 연구가 활발히 진행되고 있다. 임베디드 시스템의 가상 프로토타이핑은 제품의 디자인 개발 주기를 단축시키고 소프트웨어와 하드웨어의 통합 디자인과 시뮬레이션을 통하여 성능이 우수한 제품의 생산을 목표로 한다.

## 3. Platform 기반의 가상프로토타이핑

플랫폼은 선택사항에 따라 쉽게 플랫폼의 구성 및 변경이 가능하고, 프로세서와 시스템 프로토콜로 구성되어 프로세서의 종류, 개수, 버스구조와 계층, 데이터 종류까지 구성하는 개방형과 응용모듈이 부가되어 약간의 변형 혹은 기능 추가를 위한 기본 디자인 형이 있다. 또한 모듈별 설계를 통한 검증과 하드웨어와 소프트웨어 모듈을 동시에 개발 가능한 환경을 제공한다.

### 3.1 임베디드 시스템 개발을 위한 Platform개요

임베디드 개발도구 시장은 급속히 확대되는 추세에 있으나 상용 임베디드 OS 공급자들은 자사의 OS에 최적화된 임베디드 IDE를 번들로 출시함으로써 독자적으로 개발된 임베디드 SW개발도구의 시장의 활성화의 장애요인이 된다. 이러한 플랫폼 기반의 임베디드 시스템 개발 도구는 SoC(System on Chip) 개발에 적용이 급속도로 증가하고 있다. 임베디드 시스템 응용프로그램의 검증을 위하여 상위수준의 언어(C, SystemC)로 기술되며, 목표 플랫폼에 따라 응용프로그램을 하드웨어와 소프트웨어로 구현될 영역을 구분한다. 변경 가능성이 있는 부분은 소프트웨어

로 구현되고 따라서 최적화된 프로세서와 OS가 제공된다. 라이브러리에 등록된 플랫폼은 프로세서와 하드웨어 모듈, IP(Intellectual Property) 등이 통합되어 하드웨어와 소프트웨어의 co-design과 co-simulation이 수행된 후 FPGA 혹은 전용 에뮬레이터를 통한 최종 검증(verification)이 수행된다.

### 3.2 Platform기반의 가상프로토타이핑 개발도구 특징

임베디드 시스템에서 주어진 시스템의 기능을 하나의 칩에 구현하기 위한 SoC의 개발에 사용이 확대되고 있다. 하나의 칩에 시스템을 완성하기 위하여 제조공정이 다른 여러 기능 블록들을 집적하는데 주요 블록은 다음과 같다.

- Embedded Microprocessor
- 메모리
- 외부 시스템과 통신을 위한 장치
- 데이터 전송 블록
- Analog, RF, MEMS 블록 등

플랫폼 기반의 임베디드 시스템 개발 툴 역시 객체지향적 개발방법론에 기반한 IP의 재사용을 통하여 시스템의 자원을 활용하고 IP간의 연결과 버전 문제를 조율함으로써 어플리케이션 입장에서 시스템을 모니터링 할 수 있는 능력과 시스템이 사용할 IP 선정 능력을 동시에 지닐 수 있다.

이러한 플랫폼 기반의 개발도구는 어플리케이션의 변화에 대한 수용 가능한 일반적 구조 여부가 포인트가 되며, 세 분야의 세부 플랫폼으로 구성되어 검증 영역을 포함하는 플랫폼을 제공한다.

- 하드웨어 플랫폼  
programmable core, I/O subsystem Memory
- 소프트웨어 플랫폼  
커널을 포함하는 OS, 컴포넌트 IP에 대한 디바이스 드라이버 등
- 시스템 플랫폼  
앞서 소개된 두 플랫폼의 공통점을 찾아 시스템을 구현하기 위한 플랫폼
- 각 플랫폼을 미리 만들어 API에 등록된 하드웨어와 소프트웨어의 속성에 따라 시스템을 구현하고 시

물레이션을 통한 검증기능까지 가능하다.

플랫폼 기반의 개발도구는 하드웨어와 소프트웨어 및 시스템 구성 요소들을 사용시 최적화 된 상태로 API에 포함하여 사용자의 요구에 따라 선택 가능하며 각 컴포넌트는 재사용을 통한 편리한 임베디드 시스템의 구현 및 디버깅, 시뮬레이션과 검증테스트가 가능하다.

### 3.3 Platform 기반 개발도구의 종류

Virtio, Esto, Ptolemy 등의 툴이 있으며, 이중 특히 Virtio의 경우 Virtual Platform에 기반하여 SoC의 가상 프로토타이핑 환경을 제공한다.

플랫폼 기반의 개발도구는 편리한 GUI 환경의 제공 및 컴포넌트의 재사용을 통한 작업량의 획기적 감소에도 불구하고 실제 사용은 미미하다.

본 논문에서는 플랫폼 기반의 가상프로토타이핑으로 Virtio를 Intel의 Xscale PXA255 기반의 칩셋과 관련된 가상플랫폼 기반으로 구현하기 위하여 PXA255 칩셋 중 UART 입출력을 프로토타이핑 한다.

## 4. Virtio 개요

제품의 개발 주기 단축을 목표로 초기 개발 단계에서 완벽한 임베디드 시스템의 소프트웨어 모델을 제공함으로써 신속한 임베디드 소프트웨어 개발을 위해 플랫폼에 맞는 소프트웨어 모형을 제작, 실제 플랫폼 기반의 개발환경을 제공한다.

### 4. PXA255 칩셋 중 UART 입출력을 프로토타이핑

Intel의 Xscale PXA255 기반의 칩셋을 가상플랫폼 기반으로 구현하기 위하여 PXA255 칩셋 중 UART 입출력을 프로토타이핑 한다. 이를 위한 시뮬레이션 스케줄링은 기본적으로 시분할 스케줄링으로 구현하며, VRE(Virtio Runtime Environment) 마스터들은 각각 라운드로빈 스케줄링 알고리즘으로 실행한다. UART 입출력 인터페이스는 VSP(Virtual Serial Port)를 사용한다.

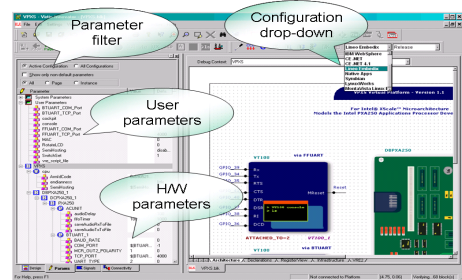
#### 4.1 구현 시스템

RAM : 512M  
 OS : Windows 2000 Professional  
 개발 도구 : Virtio VPXS platform

#### 4.2 구현방법

##### 4.2.1 하드웨어 및 소프트웨어 디자인

임베디드 시스템의 개발, 디버깅, 실행의 가상 컴포넌트와 플랫폼을 제공하는 Virtio innovator는 Microsoft의 Visual C++ 컴파일러와 유사한 인터페이스로 제공된다. 툴 컴포넌트는 디자인 브라우저와 Graphical Magic-C 에디터와 디버거, Code Generator를 통하여 C++코드가 생성되며 Plug-and-play로 테스트 할 수 있는 Test Bench Editor로 구성되어 있다. <그림 1>은 Innovator로 하드웨어를 구성하는 과정을 보여주고 있다.



<그림 1> 하드웨어 모델링

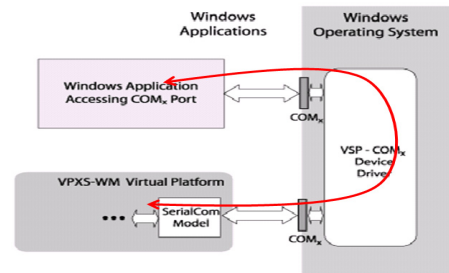
UART는 하나의 명령을 처리하기 위한 블록 컴포넌트로 PXA255 프로세서는 3개의 UART를 포함하는데 BlueTooth UART(BTUART), Full Function UART(FFUART)와 Standard UART(STUART)로 구성된다.

가상 VPXS-WM 플랫폼은 시리얼 포트와의 채널 인터페이스를 제공하며 데이터 전송은 윈도우 API 함수로 실행한다.

##### 4.2.2 시스템 입출력 인터페이스 설계

Serial COM port와 인터페이스를 구현하기 위한 가상 포트로 VSP(Virtual Serial Port)를 사용한다.

VSP는 DDI(Device Driver Interfaces) 표준 윈도우의 통신 포트(COM port)를 표시하는 것으로, UART,Modem과 같은 실제 하드웨어를 통하여 수행되는 데이터로 다른 어플리케이션이 VSP로 액세스를 허용하는 I/O 인터페이스의 설계가 가능하다.

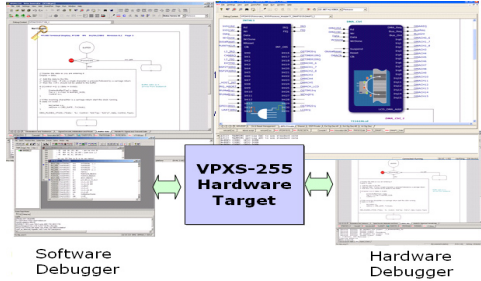


<그림 2> VSP 동작 흐름

##### 4.2.3 하드웨어와 소프트웨어 통합 디버깅

<그림 3>은 통합 디버깅 환경을 보여준다. 디버깅

은 VRE로 시뮬레이션, 컴포넌트의 로딩과 스케줄링 통합 관리 및 VDI(Virtio Debugger Interfaces)를 제공한다. 또한 VRE는 ISSs, SystemC, C로 변환된 HDL등과 같은 써드파티 디버거와 쉽게 통합이 가능하게 하는 역할을 한다.



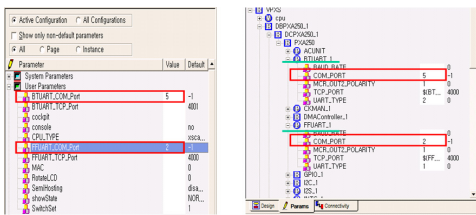
<그림 3> Virtio Debugger Interfaces

## 6. 시스템 구현

### 6.1 내부 구현

PXS255 보드의 3개 UART중 FFUART와 BTUART의 포트를 각각 2와 5로 셋팅하고 BAUD를 각각 19200으로 설정하였다.

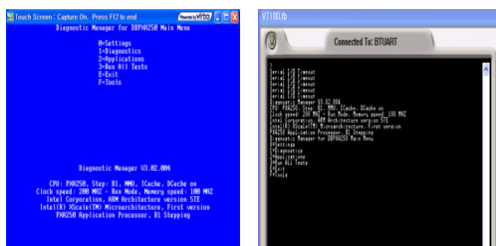
다음은 파라미터값을 설정하기 위한 호스트와 타겟의 값을 설정한다.



<그림 5> VSP 설정

각 Port의 값을 5와 2로 설정하고 우측의 실제 포트를 각각 VSP와 같은 값의 포트로 설정하였다.

<그림 6>과 UART의 가상프로토타이핑을 VSP 없이 실행한 결과를 보여준다.



<그림 6> VSP 없는 UART I/O 인터페이스

I/O 및 각각의 인터페이스는 VSP를 사용하지 않아도 무관하나, 가상 플랫폼에서 가상의 시리얼포트의 연결에서 VSP는 임베디드 칩셋의 가상 프로토타이핑 및 임베디드 컴퓨팅에 적용이 가능하다.

## 5. 결론 및 향후 과제

Virtio는 Intel 및 ARM사의 다양한 마이크로프로세서기반의 개발환경 및 IP의 구현을 위한 다양한 플랫폼을 제공한다. 하지만, 플랫폼 기반의 개발도구들이 앞서 기술한 바와 같이 많은 편리함을 제공하고 있으나 실제 제품개발현장에서는 활용도가 낮다. 이는 하드웨어와 소프트웨어를 동시에 설계하고, 시뮬레이션으로 검증이 되었다고는 하나 실제 하드웨어를 설계 및 검증 테스트 결과의 차이에 대한 불안감으로 가상프로토타입이 실제 하드웨어와의 연동과정에서 신뢰성이 검증된 개발도구가 없기 때문이라고 분석한다.

따라서 각 개발하고자 하는 목표시스템의 도메인을 만족하는 신뢰성있는 개발 도구는 복잡하고 고난도의 기술을 요구하는 임베디드 제품개발시 필요한 도메인의 특성에 따른 부분만 사용함으로써 개발자들의 편리함과 실제 제품이 출시 되었을때 제품의 오류를 줄이는 획기적 방법이 될 것이다.

향후 실물프로토타입과 가상프로토타입의 실시간 연동을 위한 인터페이스의 개발은 임베디드 시장의 기술선점과 제품의 경쟁력 강화로 이어질 것이다.

### [참고문헌]

- [1] 이정배 등, “국내 임베디드 S/W산업 실태조사에 관한 연구”, 한국소프트웨어진흥원, 2004.3.  
[http://iita6.iita.re.kr:8888/korean/journal/13/focus\\_01.htm](http://iita6.iita.re.kr:8888/korean/journal/13/focus_01.htm), 2002. 9.
- [3] 정보통신산업 중장기 시장전망, KISDI, 2000.
- [4] 2002 Embedded Software Tools Worldwide Forecast, Gartner Dataquest Market Statistics 110850, 2002. 10.
- [5] Worldwide Embedded Software Tools Outlook, 2002, Gartner Dataquest Alert, 2002. 10.
- [6] The 2000 Embedded Software Strategic Market Intelligence Program : Embedded Operating System, Software Development Tools, and Desing Automation Tools, VDC, 2000.
- [10] 김홍남, “유비쿼터스 컴퓨팅 기기들을 위한 임베디드 S/W 플랫폼 기술 개발 계획”, 정보처리학회지, 제 10권 제4호, 2003. 7. pp.39~48
- [11] Brown, A., Large-Scale, Component-Development, Prentice Hall, 2000.
- [12] <http://www.e-sim.com>
- [14] <http://www.embeddedworld.co.kr>