

RAPID 기반의 통합개발환경 인터페이스 설계 및 구현

이정배* · 서일수**

요 약

본 논문에서 RAPID를 이용한 통합개발환경을 위하여 가상 프로토타이핑 기반의 통합개발환경 연동 인터페이스를 설계 및 구현하였다. 연동 인터페이스를 통하여 서로 다른 가상 및 실물 임베디드시스템 프로토타입들간의 통합이 가능함을 제시하였다. 특히 구현 결과의 시험을 통하여 연동 인터페이스의 우수함을 입증하였다.

Design and Implementation for Integrated Development Environment Interface Based on RAPID

Jeong Bae Lee* · Il Soo Seo**

ABSTRACT

In this paper, Integrated development environments interface was designed and implemented for the integrated development environments. By using connection interface, Integration between physical prototyping and virtual prototyping which has different characteristics each other could be possible. Specially, good performance of the connection interface was showed by testing result of operation implemented.

Key words : RAPID, Physical Prototyping, Virtual Prototyping, Embedded System

접수일 : 2009년 4월 15일; 채택일 : 2009년 5월 20일

* 선문대학교 컴퓨터공학부

** 대구대학교 컴퓨터·IT공학부

1. 서 론

임베디드시스템은 특수한 목적을 위하여 사용되는 제품에 내장되어 사용자 중심의 서비스를 제공할 수 있다. 많은 산업 분야와 실생활에 사용되는 정보 가전 분야 전반에 걸쳐 임베디드 기반 시스템이 널리 이용되고 있다.

이러한 관점에서 볼 때 제품의 경쟁력 확보를 위해서는, 개발 난이도가 높더라도 신제품 개발에 있어서 고객의 요구를 적극 수용할 수 있어야 하고, 신제품이 시장에 빠르게 진입할 수 있는 개발 환경이 보장되어야 한다.

이러한 문제점들을 해결하기 위하여 개발 시간을 단축하기 위한 방법을 찾게 되었고, 이 과정에서 일반적으로 적용되는 기법이 프로토타이핑이라 할 수 있다. 임베디드시스템 개발에 사용된 프로토타이핑은 가상 프로토타이핑, 실물 프로토타이핑, 통합 프로토타이핑의 세 가지 방법론이 사용된다. 실물/가상 프로토타이핑의 장점을 보완하여 만들어진 기존 통합 프로토타이핑 시스템은, 소켓통신의 연결 인터페이스를 이용하여 이미 만들어진 가상 프로토타이핑과 실물 프로토타이핑을 통합하여 실물/가상 프로토타이핑에 대한 동시 시뮬레이션 환경을 제공하였다. 기존 통합 프로토타이핑 시스템은 가상 프로토타이핑의 장점 중 하나인 시스템 요구 및 제약 조건을 추출하여 사용자의 요구 사항을 시각적으로 바로 보여줄 수 있는 부분이 희생 되었으며, 사용자의 요구사항에 대한 분석과 개발자간의 의사소통이 마무리되어 실제적인 임베디드 제품 개발과 테스트 시에는 기존 시스템을 이용할 수 없다는 문제가 있다.

본 논문에서는 고객 요구사항에 대한 신속한 분석과 분석 후 빠른 개발을 위한 방법으로 가상 프로토타이핑 개발 도구(RapidPLUS)를 이용한 통합 개발환경 인터페이스를 제시한다.

통합개발환경은 실제 소프트웨어가 실행되는 시스템(타겟)과 개발하는 시스템(호스트)이 다른 개

발환경을 말한다. 임베디드시스템의 특수성으로 인해서 타겟에서 S/W를 직접 개발하는 것은 어려우므로 보통 통합개발환경을 구성하게 된다. 통합개발환경을 구성한 후 시스템을 호스트에서 개발하고, 크로스 컴파일러(Cross Compiler)를 이용하여 타겟에 의존적인 실행코드를 생성하며, 그 실행코드를 타겟에 다운로드 및 이식하여 실행할 수 있는 필요성이 요구된다. 이를 위하여 가상 프로토타이핑 기반의 통합개발환경 연동 인터페이스를 설계 및 구현하였으며, 구현된 연동 인터페이스를 통하여 서로 다른 가상 및 실물 임베디드시스템에서 프로토타입들간의 연동이 이루어진다. 새로운 연동 인터페이스를 통하여 임베디드시스템의 개발 기간 단축과 재사용성 및 확장성을 크게 높일 수 있다.

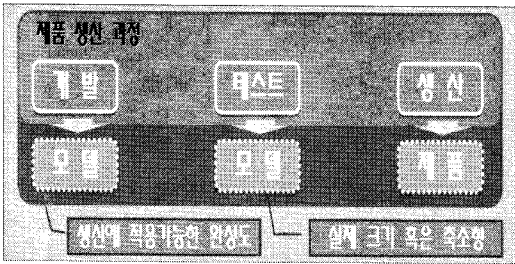
2. 관련 연구

본 장에서는 임베디드시스템 개발에 있어서 기존 사용되고 있는 실물 프로토타이핑, 가상 프로토타이핑, 통합 프로토타이핑 기술과 임베디드시스템 개발도구들을 소개하도록 한다.

2.1 프로토타입과 프로토타이핑

프로토타입(Prototype)이란 제품 생산 전에 외형 및 기능을 시험하기 위하여 만든 실제 형태의 실물 모델이다. 프로토타입을 만드는 이유는 신제품을 출시하기 전에 소비자의 요구를 제품 개발에 적극 반영하고, 오류를 찾아내어 신뢰성과 경쟁력 있는 제품을 개발하기 위함이다.

(그림 1)에서와 같이 제품 개발 단계에서 개발 제품의 모형 중 생산에 적용 가능한 완성도 있는 모델을 만들어 내는 것을 프로토타이핑이라하며, 또한 테스트 단계에서 제품 생산 전에 외형 및 기능을 시험하기 위하여 만든 실제 크기 혹은 축소형의 실물 모델이 프로토타입이다.



(그림 1) 프로토타입의 단계별 의미

프로토타이핑(Prototyping)이란 프로토타입을 컴퓨터 기술을 이용하여 구현하는 것으로, 3차원 객체의 형상을 시각화하고 정의하는 수단을 제공하며, 최첨단 기술을 적용한 임베디드시스템 설계 시 다양하게 응용되고 있다. 또한 제품의 수명주기가 점차 짧아지고, 제품의 경쟁력이 개발기술과 더불어 출시 일정에 따라 좌우될 가능성이 충분하다. 특히 임베디드 기술이 적용된 제품의 경우 새로운 설계 및 요구분석을 위한 최적의 프로토타이핑 방법론의 개발과 적용은 신제품의 출시 일정을 결정하는 가장 중요한 요소로 작용한다. 따라서 새로운 프로토타이핑 방법론의 필요성을 간단히 살펴본다.

① 프로토타이핑 대상의 속성 변화

기존의 프로토타이핑은 정적이고 하드웨어 중심으로 진행되어 왔다. 따라서 제품 개발과정에서 크기와 무게만 고려하여 제작되는 형태에서 컴퓨터 기술의 발전으로 CAD(Computer Aided Design)나 3차원 입체 도구와 같은 형태로 발전되었으나 여전히 동적인 프로토타이핑보다는 외형의 디자인에 중점을 두었다. 하지만, 최첨단 기술의 발달로 복잡하고 큰 시스템 구현 시 효율적인 제품 개발방법으로 프로토타이핑 방법의 적용이 확대됨으로써 동적이고 지능적으로 상호작용 가능한 프로토타이핑 방법론이 요구되었다.

② 프로토타이핑 대상의 다양화

고전적인 프로토타이핑은 제품의 외형과 시각

적 요소를 고려하였으나 점차 전 산업 분야에 적용되면서 하드웨어, 소프트웨어 및 행위 프로토타이핑과 사용자 중심의 상호작용이 가능한 체험적 서비스를 제공하는 등의 다양한 분야에 적용이 확대됨으로써 각 분야의 특수성에 맞는 프로토타이핑 방법론이 필요하게 되었다.

이러한 프로토타입의 특징을 반영하는 바람직한 프로토타이핑은 경제성, 사용 편의성, 접근 용이성 등의 기준을 만족하는 기법으로 발전하고 있으며, 새로운 프로토타이핑 기법이 활발히 연구되고 있다.

현재까지 연구된 프로토타이핑의 유형은 많은 종류가 있으나 한 두 가지의 프로토타입으로 설계상의 모든 문제를 모두 해결할 수 없으며, 각 프로토타입의 핵심적인 특성을 감안하여 여러 프로토타입을 혼용하여 개발할 필요가 있다.

최근 복잡하고 고난도의 개발기술을 요구하는 임베디드시스템 분야에서 신뢰성 있는 제품의 신속한 개발 생산을 위하여 신속 프로토타이핑(Rapid Prototyping) 방법론이 활발하게 적용되고 있으며, 실제적 유형으로는 (그림 2)와 같이 가상 프로토타이핑, 실물 프로토타이핑, 통합 프로토타이핑의 세 가지로 구분된다. 프로토타이핑은 설계 대상에 따라 다양한 프로토타이핑이 적용되며, 전문적이고



(그림 2) 프로토타이핑의 종류

긴밀한 협업을 통하여 시장 진입 시점(Time-To-Market)이 단축되는 효과를 얻을 수 있기 때문에 현재 가상 프로토타이핑과 실물 프로토타이핑을 위한 여러 가지 도구가 개발되고 있다.

본 절에서는 가상 프로토타이핑, 실물 프로토타이핑, 통합 프로토타이핑 기술 종류에 대해서 간략하게 소개한다.

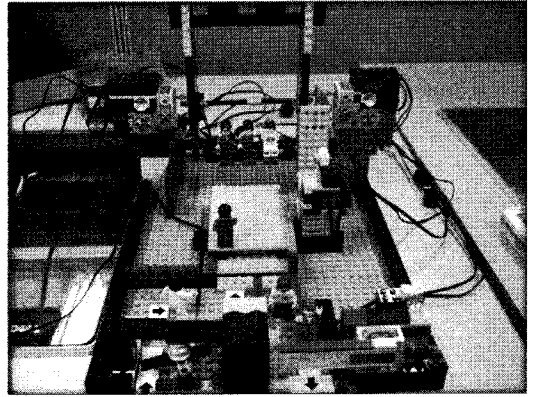
2.2 실물 프로토타이핑

실물 프로토타이핑(Physical Prototyping)은 실제 품과 같은 외형과 제품의 기능을 탑재한 실제 크기 혹은 축소형의 실물 모델을, 임베디드시스템에 공통적으로 활용되는 표준 센서와 액추에이터, 그리고 쉽게 조립할 수 있고 재사용 가능한 부품 및 블록들을 활용하여 실제 개발하는 것이다. 이러한 개발 방법은 제품의 기획, 외관 및 제품의 기능설계와 모형 제작 후 직접 테스트 및 제어를 통하여 제품의 외형이나 오류를 수정하고, 사용자의 요구를 반영한 시제품 검증을 수행하기 때문에 출시 이후에 제품 결함으로 인한 문제점을 현저히 줄일 수 있다.

또한 모형제작에 따른 시간적 경제적 부담을 줄이면서 제품의 신뢰도를 높일 수 있고, 임베디드시스템 특성상 공동 작업을 수행하는 다양한 전문성들의 이해증진을 위한 도구로도 활용되며, 최적의 문제 해결 방법을 찾는 개발 도구로의 활용까지 가능하다.

그러나 하드웨어 개발이 완료된 다음에 소프트웨어 개발이 진행되기 때문에 개발 인력에 대한 효과적인 활용이 불가능하고, 이로 인해 개발 기간이 길어지며, 빠르게 증가하는 제품의 복잡도에 따라 임베디드시스템 제품 설계에서 검증까지 필요한 시간과 비용의 부담으로 시장 경쟁력을 상실하며, 제품의 외형이 바뀌면 새로운 프로토타입을 만들어야 하고, 소형 임베디드 제품에 대한 시뮬레이션이 어렵다는 단점을 가지고 있다.

(그림 3)은 실물 프로토타이핑 모형으로 레고로 구현한 사진을 표현하고 있다.



(그림 3) 실물 프로토타이핑

최근 실물 프로토타이핑을 이용하면 개발 과정에서 하드웨어 개발자와 소프트웨어 개발자가 동시 작업을 진행하면서, 소프트웨어를 적용할 때 발생할 수 있는 문제점들을 바로 하드웨어 개발자에게 전달하게 된다. 따라서 최종 품질을 개발 단계에서부터 보장할 수 있는 등 실물 프로토타이핑의 중요성이 강조되면서 개발 도구들이 활발히 연구 개발되고 있는데, 경북대학교 Real-Time System 연구팀에서 개발한 ESPS(Embedded System Prototyping Suit)가 대표적인 예이다.

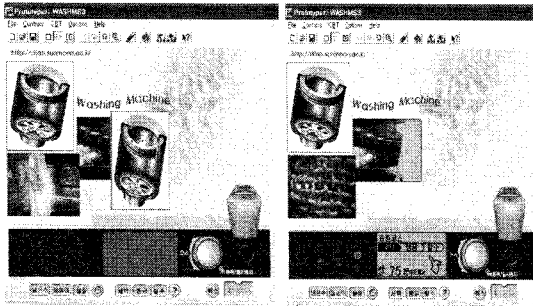
2.3 가상 프로토타이핑

가상 프로토타이핑(Virtual Prototyping)은 임베디드 소프트웨어 개발 초기단계에서 사용자 혹은 의뢰자의 요구사항들을 효과적으로 추출할 수 있도록 고안된 시스템 요구 및 제약 조건 추출 방법을 말한다. 특히 컴퓨터 프로그램을 이용하면 가상 프로토타입을 사용자에게 시연하면서 사용자의 요구 사항에 맞도록 모양을 바꾸거나 새로운 기능을 추가하는 등의 변화를 쉽게 반영할 수 있다. 현재 다양한 목적으로 개발되는 각각의 임베디드시스템을 내장하여 출시되는 제품은 최첨단의 기술과 기능을 요구하는 디지털 산업 분야가 대부분이다. 가상 프로토타이핑 개발방법의 적용은 복잡

한 인터페이스를 가진 제품의 개발 시간을 단축하게 되어 출시된 제품의 수명 주기도 점차 짧아지게 하고 있다.

가상 프로토타이핑은 컴포넌트와 라이브러리를 제공함으로써 정밀한 단위까지 시뮬레이션이 가능하고, 실시간으로 변경된 디자인을 관찰함으로써 소비자에게 친근감이 가는 디자인으로 변경이 용이하다.

그러나 실제품을 시뮬레이션 하기 위한 구동환경을 제공하지 못하고, 컴퓨터 기술을 이용한 가상 시뮬레이션으로 생길 수 있는 오류/에러로 인한 추가적인 부담이 발생할 수도 있다.



(그림 4) 가상 프로토타이핑

(그림 4)는 가상 프로토타이핑 제작 도구인 RapidPLUS를 가지고 세탁기를 구현한 사진이다.

가상 프로토타이핑 기술을 적용한 사례는 가상 공장 구축 시스템, 공항 관제 시스템의 설계, 건설 회사의 모형 아파트(모델 하우스) 설계, 자동차 모형 개발과 같은 큰 규모의 프로토타입에서부터, 가전제품 개발, 카오디오 설계와 같은 소형 프로토타입 검증 분야에 이르기까지 광범위하고 다양하다. 향후 대부분의 생산 분야에서 가상 프로토타이핑을 통한 제품 개발이 주류를 이룰 것으로 전망되고 있다.

현재 가상 프로토타이핑 개발도구로는 e-sim사의 RapidPLUS, 포항공대에서 개발한 ASADAL, RT-LAB과 UML(Unified Modeling Language)

등 통합 개발 방법론과 결합된 CASE(Computer-Aided Software Engineering) 형태로, i-Logix사의 UML기반 개발도구인 Rapsody, 아이너스 기술의 PlayMo 등이 있다.

특히 가상 프로토타이핑 개발도구 PLUS는 GUI 기반의 도구로서 임베디드 개발에 필요로 한 여러 가지 Object들을 컴포넌트화 하고 있다. 따라서 임베디드 가상 프로토타이핑 작업 시의 편의성은 물론 Active X, Java Bean등을 지원하기 때문에 기존 Basic, C, Java로 구현한 컴포넌트 및 라이브러리 등을 사용할 수 있다.

2.4 통합 프로토타이핑

실물/가상 프로토타이핑은 실물 프로토타이핑과 가상 프로토타이핑을 통합한 설계 방법으로, 제품 출시 전에 외형 및 성능 시험을 위한 테스트 모델을 제작하는 것을 말한다.

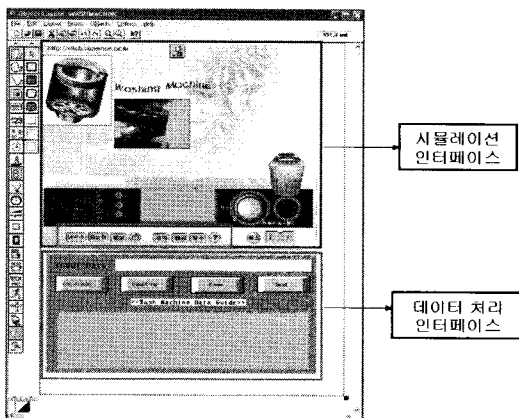
가상 프로토타이핑은 실제품을 소프트웨어로 구현하기 때문에 유동성 있는 시스템의 프로토타입을 제작할 수 있지만, 실제품 적용에 대한 신뢰에 대한 문제가 제기된다. 실물 프로토타이핑은 실제품에 대한 신뢰도는 좋지만 개발 환경에 크게 영향을 받고, 세부적 구현에 많은 어려움이 존재한다.

앞서 제시한 Real-Time Systems 연구팀에서 개발한 ESPS를 활용하여 제작된 실물 프로토타입과 연동되는 기존 통합 프로토타이핑은, 실물/가상 프로토타이핑에 대한 동시 시뮬레이션으로 임베디드시스템 제품의 신뢰성을 보장하고 있다. 동시 시뮬레이션을 통하여 제품 개발에 대한 시간과 비용이 감소함에 따라 시장 진입 시점이 빨라지고, 시장 경쟁력의 우위 효과도 얻을 수 있다.

통합 프로토타이핑(Integrated Prototyping)은 실물 프로토타이핑과 가상 프로토타이핑의 연동으로 구현이 되는데, 가상 프로토타이핑의 융통성 및 가변성을 이용하여 시스템의 유저 인터페이스 및 세부적인 기능을 표현하고, 실물 프로토타이핑의 직관성을 이용하여 시스템의 큰 동작 및 외형을 표

현한다. 특히 연결 인터페이스와 동시 개발이 가능하다. 그리고 TCP/IP 소켓과 ActiveX를 이용하여 정보가전 시물레이션 환경을 구축하였으며, 가상 프로토타이핑과 실물 프로토타이핑이 주어진 조건에 맞게 연동되어 향후 많은 임베디드시스템에서 적용 가능함을 보이고 있다.

(그림 5)는 통합 프로토타이핑을 구현한 사진을 표현하고 있다.



(그림 5) 통합 프로토타이핑

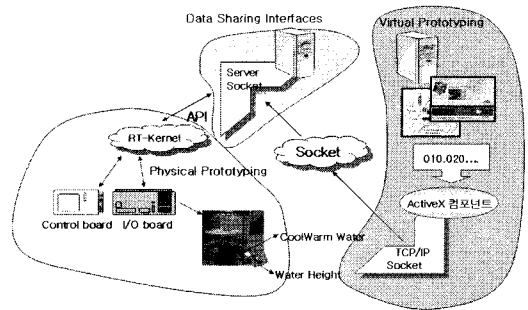
현재 가상 프로토타이핑과 실물 프로토타이핑을 위한 여러 가지 도구가 개발되고 있으며, 보다 신뢰성 있는 제품의 개발을 위한 획기적이고 새로운 개발 방법론의 제시가 이루어지고 있다. 제시되는 모델들은 시장 진입 시점(Time-To-Market)의 단축을 가능하게 하는 새로운 임베디드시스템 프로토타이핑 방법으로 정착되고 있다.

그러나 현재까지 통합 프로토타이핑은 소형 임베디드시스템 소프트웨어 개발에 적합하지 않으며, 네트워크 연결이 이루어지지 않으면 사용할 수 없다는 단점을 가지고 있다.

현재까지 연구된 통합 프로토타이핑 개발 방법론은 앞서 살펴본 실물 프로토타이핑과 가상 프로토타이핑의 연동을 위한 인터페이스만 제안이 되어 있는 실정이다. (그림 6)은 인터페이스를 통해

서 가상 프로토타이핑과 실물 프로토타이핑이 연동 가능함을 보여준다.

이렇게 구현되어 있는 통합 프로토타이핑 시스템은 사용되어지는 모든 센서나 액추에이터에 대해서 각각의 기능과 동작을 개별적으로 구현하므로 효율성이 낮은 편이다.



(그림 6) 통합 프로토타이핑 연동 시스템의 구성

3. 시스템설계 및 구현

3.1 직렬(Serial) 통신 연결

임베디드시스템 개발 장비(X-Hyper 320TKU)에는 (그림 7)과 같이 시리얼 케이블로 DEBUG1이라고 표시되어 시리얼 커넥터와 호스트 PC의 직렬(COM1/2)포트에 연결해 주면 된다.



(그림 7) 타겟 보드와의 Serial 연결

3.2 TCP/IP(Ethernet) 소켓 연결

이더넷 크로스 케이블(Ethernet cross cable)을 (그림 8)과 같이 연결한다. 임베디드시스템 개발 장비(X-Hyper320TKU)의 이더넷 포트와 Host PC의 RJ45(이더넷 카드)를 제공한 크로스 LAN 케이블로 연결한다. 만약 이더넷 허브에 직접 연결하려면 일반 Direct LAN cable로 연결이 가능하다.



(그림 8) 타겟 보드와의 TCP/IP(Ethernet) 소켓 연결

```

while(1)
{
  if(pthread_create(&client_tid, NULL, client_thread, NULL) != 0)
  error_handling("thread() error");
  if (str_len=recv(client_sock,c,sizeof(c),0) == -1)
  {
    error_handling("recv error");
    exit(1);
  }
  switch (c[0])
  {
    case '1': char_cd(streamParsing(c,1));
              break;
    case '2': segment(streamParsing(c,1));
              break;
    case '3': segPos = c[1]-48;
              segmentAt(segPos,streamParsing(c,2));
              break;
    case '4': segPos = c[1]-48;
              do(segPos);
              break;
    case '5': segPos = c[1]-48;
              setLed(segPos);
              break;
  }
  printf ("Msg : %s\n",c);
  send(client_sock,c,sizeof(c),0); /* echo */
  printf ("ok\n");
  sleep(1);
  memset(c, 0x00, sizeof(c));
}
char strTemp[BUF_SIZE];
char *streamParsing(char *str, int cpleng)
{
  int i;
  memset(strTemp, 0x00, sizeof(strTemp));
  strcpy(strTemp,str);
  for (i=cpleng;i<sizeof(strTemp);i++)
  {
    strTemp[i-cpleng] = strTemp[i];
  }
  return (char *)strTemp;
}
    
```

(그림 9) 임베디드시스템 개발 보드 서버

디바이스 드라이브와 API를 이용하면 (그림 9)에서 보듯이 응용프로그램과 임베디드시스템 개발 보드가 TCP/IP 소켓으로 연결될 수 있는 엔진으로 구현 및 활용이 가능하다.

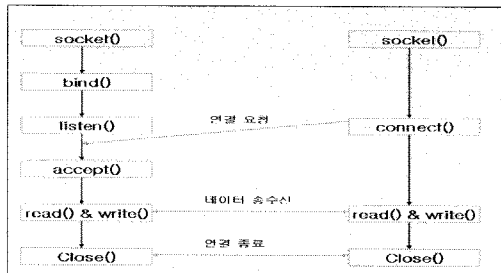
가상 프로토타이핑 개발도구 컴포넌트에는 임베디드시스템 개발 보드의 디바이스 드라이버에 맞는 변수 값이 할당되어 있다. xhyper320 TKU_Socket 컴포넌트에 있는 connect() 메소드를 이용해서 임베디드시스템 개발 보드 Server로 접속한 후 send[디바이스]() 메소드를 호출하게 되면, Server에서는 streamParsing() 함수를 통해서 가상 프로토타이핑 개발도구에서 넘어온 디바이스 값을 확인 한 후 쓰레드를 생성시킨다. 생성된 쓰레드에서 해당 디바이스에 대한 명령이 실행되도록 하였다.

3.3 TCP/IP 서버 소켓 설계

가상 프로토타이핑 모듈에서 전달되는 데이터를 소켓을 통하여 수신하기 위한 서버를 설계한다. 소켓을 이용한 클라이언트와 서버와의 통신을 위하여 아래와 같은 흐름에 따라 함수가 호출된다.

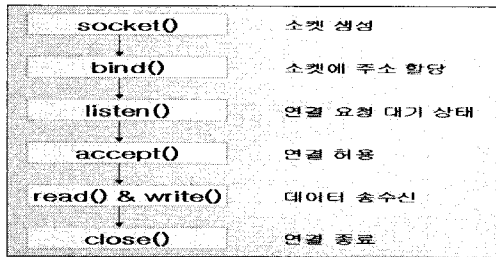
- ① socket()을 사용하여 TCP 소켓 생성
- ② bind()로 소켓에 한 포트 번호를 부여
- ③ 시스템에 listen()을 생성된 포트에 연결요구를 허락하도록 알림
- ④ 다음과 같은 과정을 반복

▶ (그림 10)과 같이 각 클라이언트 연결에 대해 새로운 소켓 생성을 위한 accept() 호출



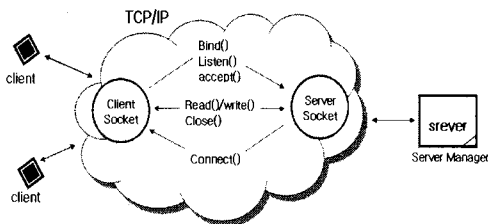
(그림 10) 클라이언트 및 서버 함수 호출

- ▶ read()와 write()로 생성된 소켓을 통해 클라이언트와 통신
- ▶ (그림 11)와 같이 close()로 해당 클라이언트와 연결 종료



(그림 11) 서버 함수 호출

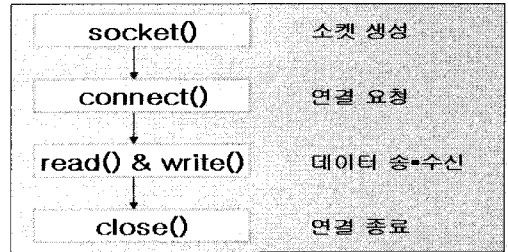
(그림 12)는 TCP/IP의 클라이언트와 서버 함수 호출관계를 그림으로 나타내었다.



(그림 12) TCP/IP 클라이언트와 서버 함수 호출 관계

3.4 TCP/IP 클라이언트 소켓 설계

본 시스템에서는 서로 멀리 떨어져 있는 호스트들이 서로 데이터를 주고 받을 수 있도록 하며, 소프트웨어 차원에서 추상적인 장치 역할을 하는 소켓을 사용한다. 소켓에서는 클라이언트/서버라고 불리는 통신 당사자가 있다. 클라이언트 프로그램은 통신을 시작하고, 서버 프로그램은 수동적으로 기다리면서 자신에게 접근한 클라이언트들의 요구에 응답한다. 본 절에서는 데이터 연동을 위한 클라이언트 소켓을 설계한다. 클라이언트는 다음 (그림 13)과 같이 4단계를 수행한다.

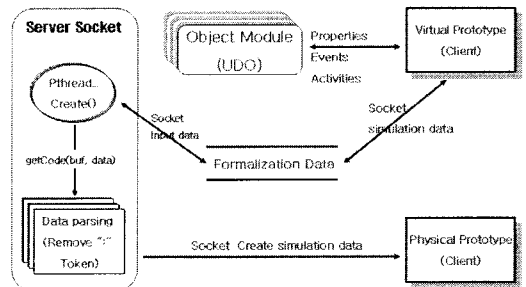


(그림 13) 클라이언트 함수 호출

3.5 데이터 명세 인터페이스 설계

가상 프로토타입 환경에서 시뮬레이션을 통한 데이터를 TCP/IP 소켓으로 전송하면, 서버 소켓에서는 수신 데이터를 실물 프로토타입 시뮬레이션을 위한 데이터로 변환하는 인터페이스 설계가 요구된다.

(그림 14)는 데이터 변환 명세를 위한 인터페이스이다.



(그림 14) 가상 및 실물 데이터 변환 인터페이스

가상 프로토타입의 연동 인터페이스에서 선택한 데이터는 서버 소켓을 통하여 전송되고, 해당 문자열을 파싱 후 앞서 정의한 데이터 명세화에 따른 실물 프로토타입 시뮬레이션을 위한 데이터로 변환되어 해당 동작을 수행한다.

(그림 15)는 데이터 처리를 위한 실제 인터페이스를 담당하는 getCode()를 의사코드(pseudocode)로 나타내었다.

getCode()로 입력되는 '101:202:304 ...'와 같이


```

인터페이스를 위한 알고리즘
입력 : 시뮬레이션이 완성된 가상 프로토타입의 데이터 문자열
출력 : 실물 프로토타입형의 구동 데이터 전송값의 출력

{
  가상 시뮬레이션된 문자열 데이터 값을 입력받아 각각의 동작을
  ""로 구분하여 배열에 저장 :

  입력 문자열의 접합을 값을 변환하기 위한 배열에 저장 :

  while(입력된 문자열을 "") { ""를 문자를 기준으로 잘라 가져옴 )
  {
    if(입력 문자가 ' '이 아니면)
    {
      정수형 값을 생성 :
      출력할 배열에 추가하여 저장 :
    }
  }

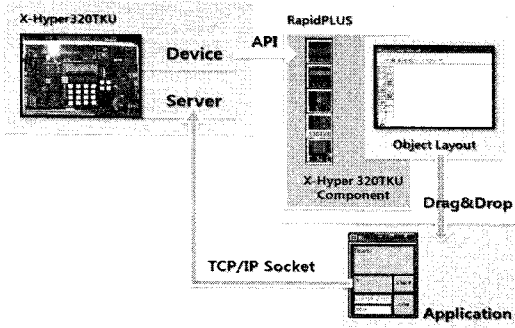
  return 출력값 :
}
    
```

(그림 15) 데이터 처리를 위한 실제 인터페이스 과정

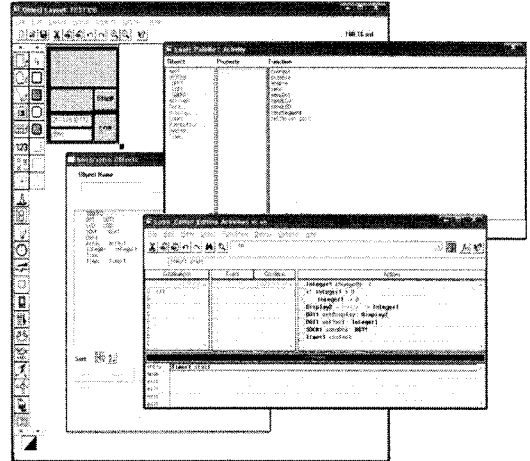
정형화된 파일로 저장된 가상 시뮬레이션 된 데이터 값이 소켓으로 연결된 스트림으로 전송된다. 전송 시 “:”문자를 기준으로 토큰이져 되어 ‘101 202 304 ...’과 같은 형태로 배열에 저장되고, 다시 앞의 데이터 ‘1 2 3 ...’이 분리되어 각각의 메뉴와 해당하는 시뮬레이션 동작으로 변환되어 실물 프로토타입이 시뮬레이션 된다.

4. 구현 및 평가

본 장에서는 제 3장에서 설계한 내용을 바탕으로 (그림 16)과 같은 구성을 갖는 임베디드시스템 개발 보드 엔진과 가상 프로토타입 개발도구 컴포넌트를 구현하였다. (그림 18)은 실제 응용프로그램을 제작하는 예를 보이고 있다.

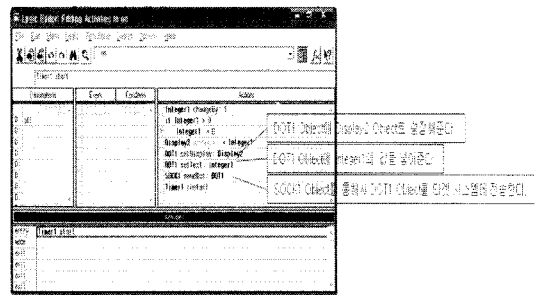


(그림 16) 시스템 구성도

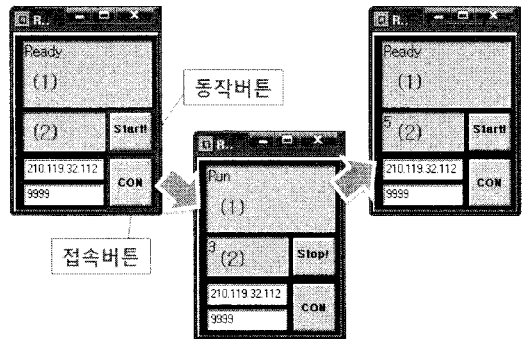


(그림 17) 응용프로그램 제작 예

(그림 18)은 오브젝트와 타겟 시스템이 매치되는 소스를 보여주고 있다.



(그림 18) 추가된 컴포넌트 사용 소스



(그림 19) 실행모습

프로그램을 제작하는 동안 프로토타이퍼를 실행시켜, 임베디드시스템 개발 보드와 연동되는지를 확인할 수 있다.

(그림 19)의 소스는 CON 버튼을 누르면 임베디드시스템 개발 보드와 소켓으로 접속이 되고, Start 버튼을 누르면 (그림 19)의 1번 디스플레이가 Ready에서 Run으로 바뀌게 되며, 2번 디스플레이는 0부터 9까지 카운터를 하는 모습을 보이고 있다. 이를 통하여 연동 인터페이스의 동작이 효율적으로 진행됨을 확인하였다.

5. 결 론

기존의 실물 프로토타이핑 방법론은 실물과 유사한 프로토타이핑을 사용해 임베디드시스템을 직접 개발하기 전에 모델링을 하는데 적합하나, 시간과 비용이 많이 소요되는 단점을 가지고 있으며, 가상 프로토타이핑은 개발 비용과 시간을 절약할 수 있다는 장점이 있는 반면에 가상 모델링 수행이 실질적인 모델링과 중복되어 개발 비용과 시간이 많이 소비된다는 단점을 가지고 있다.

본 논문에서는 통합개발환경에 적합한 가상 프로토타이핑 개발도구를 사용한 연동 인터페이스를 적용하는 방법을 구현하여 그 성능을 보였다. 먼저 통합개발환경에 적합한 가상 프로토타이핑 방법론에 대해서 제안하고, 제안한 방법론의 적합성과 우수성을 증명을 위해 단순 예제 프로그램과 인터페이스를 개발하고 이를 시험에 적용하였다.

향후 본 논문에서 소개한 통합개발환경에 적합한 가상 프로토타이핑 방법론을 타겟 시스템에도 적용할 수 있는 유저인터페이스를 제안할 필요가 있으며, 이는 실제 임베디드 제품 개발을 수행하는 도중에도 타겟 시스템에서 호스트 컴퓨터에 접근할 수 있는 유기적인 개발 환경을 제공하기 위한 과제이다.

참 고 문 헌

- [1] 정기훈, 채화영, 김정길, 이재신, 강순주, “실물 프로토타이핑 기법을 적용한 임베디드 실시간 시스템 소프트웨어 방법론”, 전자공학회지, 제31권, 제11호, pp. 30-41, 2004.
- [2] 김종일, 이정배, 양재수, 이영란, 정영진, 한강우, 강신관, 김대웅, “객체 기반의 임베디드 실물 및 가상 프로토타입 통합 기법”, 정보처리학회논문지, 제14-A권, 제4호, pp. 227-234, 2007.
- [3] 하이버스(주), 임베디드 리눅스 설계 및 응용 (PXA320 기반, Embedded Linux Kernel 2.6), 하이버스(주), 2008.
- [4] 정기훈, 채화영, 김정길, 이재신, 강순주, “임베디드시스템 프로토타이핑 기술”, 정보처리학회지, 제11권, 제6호, pp. 86-99, 2004.
- [5] 정영진, 이정배, 이영란, 권진백, 임기욱, 조상영, “프로토타이핑 기술을 적용한 임베디드시스템 가상 시뮬레이션”, 정보과학회지, 제24권, 제8호, pp. 26-39, 2006.
- [6] 이영란, “임베디드시스템의 가상 및 실물 통합 프로토타이핑 시스템 설계 및 구현에 관한 연구”, 박사학위논문, 2006.
- [7] 김종일, “임베디드시스템 개발을 위한 객체 기반의 가상 및 실물 프로토타이핑 통합 환경”, 박사학위논문, 2007.
- [8] 이정배, 김영진, 안성준, 김대웅, “자동차 임베디드 프로토타이핑 기술 소개”, 정보처리학회지, 제15권, 제5호, pp. 49-58, 2008.
- [9] (주)파이널이펙트, RepidPlus 교재, (주)파이널이펙트, 2005.
- [10] 강순주, 이정배, 박종진, 리눅스 기반 임베디드 시스템, 대영사, 2004.
- [11] 한국전자통신연구원 융합소프트웨어 연구본부, Esto3.4 사용자 매뉴얼, 임베디드 SW 기술 연구팀, 2009.
- [12] 정영진, 이정배, 이영란, 권진백, 임기욱, 조상영,

“프로토타이핑 기술을 적용한 임베디드시스템 가상 시뮬레이션”, 정보과학회지, 제24권, 제8호, pp. 26-39, 2006.

- [20] Real-Time System, <http://fsmllabs.com>.
- [21] Physical Prototyping Environment, <http://www.lego.com>.
- [22] Physical Prototyping Environment, <http://mindstorms.lego.com>.
- [23] Physical Prototyping Environment, <http://www.emulate.com>.
- [24] Virtual Prototyping Tool, “RapidPlus”, <http://www.livecomus.com>.
- [25] Linux Kernel, <http://kldp.org/KoreaDoc>.
- [26] Real-Time Linux, <http://www.rtlinux.co.kr>.
- [27] Virtual Prototyping Tool “Virtio”, Pro, <http://www.virtio.com>.



이 정 배

1981년 경북대학교 전자공학과 (공학사)
 1983년 경북대학교 전자공학과 (공학석사)
 1995년 한양대학교 컴퓨터공학부 (공학박사)

2002년 ~ 현재 선문대학교 컴퓨터공학부 교수



서 일 수

1992년 경일대학교 컴퓨터공학과 (공학사)
 1994년 경일대학교 컴퓨터공학과 (공학석사)
 2005년 대구대학교 컴퓨터정보 공학과(공학박사)

2008년 ~ 현재 대구대학교 컴퓨터·IT공학부 겸임교수