

# 산업용 시스템 진단·관리를 위한 자바 임베디드 시스템 설계 및 구현

주상현<sup>\*</sup>, 정준영, 정민수  
경남대학교 컴퓨터공학과

## Design and Implementation of Java Embedded System for Industrial Diagnosis and Management

SangHyun Joo, JunYoung Jung, MinSoo Jung  
Dept. of Computer Engineering, Kyungnam Univ.

### 요 약

현재의 다양한 산업용 대상기 시스템들을 제어, 관리 하기위한 소프트웨어들은 특정 시스템만을 위하여 존재 해왔다. 이것은 시스템의 하드웨어 종속적인 컴퓨팅 환경을 초래 하였으며, 이를 기반으로 하는 소프트웨어 개발뿐만 아니라 유지, 보수에도 많은 어려움이 뒤따르게 되었다. 본 논문에서는 하드웨어 독립적으로 소프트웨어 환경을 구성하여 이러한 어려움들을 극복하고자 한다. 따라서 산업기기와 병행하여 사용 할 수 있는 플랫폼 독립적인 자바 기반의 임베디드 시스템을 제안하고, 산업기기의 진단·관리를 위한 시스템을 구현 하였다.

### 1. 서론

오늘날 산업용 시스템들은 주어진 분야에서 주어진 작업만을 하는 시스템으로 유연성을 가지고 있지 못한다. 이러한 산업용 시스템에 유연성을 주기위해 산업용 기기와 병행 할 수 있는 임베디드 시스템을 두어서 시스템의 자동화와 무인 제어를 위해 산업 생산 시설 등에서 사용 할 수가 있다. 본 논문에서는 산업 생산 시설등에서 사용되어 질 수 있는 임베디드 시스템에 독립성과 실행환경의 통일화, 단순화 그리고 네트워크 기술을 제공해 줄 수 있는 자바 플랫폼 기반의 시스템을 제안 하였다. 자바 플랫폼 기반의 임베디드 시스템은 다양한 시스템에 적용할

수 있도록 이식성과 재구성을 높일 수 있고, 새로운 프로그램 개발의 용이성과 개발 인력에 대한 교육과 보급이 용이할 것으로 본다.

본 논문에서는 자바 플랫폼 기반의 임베디드 시스템을 바탕으로 산업 기기들의 진단·관리 할 수 있는 시스템을 구현 하였다.

### 2. 관련연구

#### 2.1 TRON

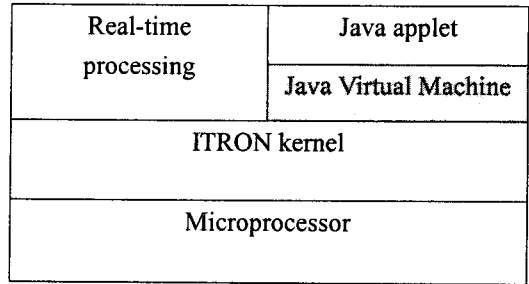
가전, 산업 기기등 인간 주변의 모든 기기에 컴퓨터를 탑재하기 위해 개발된 실시간 운영체제가

TRON(The Real-time Operating System Nucleus) 이다. TRON은 프로그램 크기가 작기 때문에 처리 속도가 빠르고, 규격을 무료로 제공하며 오픈 소스 지향이라는 장점을 가지고 있다. TRON은 ITRON, JTRON, BTRON, CTRON, TRON Human Interface의 다섯 가지 구성요소로 이루어져 있으며, ITRON과 JTRON이 핵심적인 역할을 담당하고 있다. ITRON은 임베디드 시스템을 위한 RTOS와 관련된 공개된 스펙 표준으로서, 소형화된 크기와 낮은 오버헤더를 갖는 실시간 커널이다. 광범위한 프로세서에 적용이 가능하여 다양한 임베디드 시스템에 폭 넓게 사용할 수 있다. JTRON은 자바와 ITRON을 결합한 하이브리드 OS 스펙으로서, 실시간 태스크와 자바 애플리케이션 사이의 커뮤니케이션 인터페이스에 대한 표준을 정의하고 있다. 커뮤니케이션 형태는 4가지 타입으로 정의하고 있으며, 그 타입의 기술 내용은 표 1과 같다. 타입1과 타입2는 강한 결합 멀티 프로세서 시스템이고, 타입3은 약한 결합 멀티 프로세서 시스템이다.

[표 1] 커뮤니케이션 인터페이스의 타입

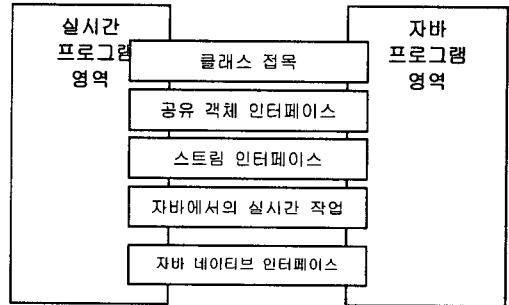
Type Name	Description
Type 1: 클래스 접목	자바 프로그램상에서 ITRON 커널 시스템 콜 사용을 허가
Type 2: 공유 객체 인터페이스	공유 객체를 통해서 통신하는 자바 프로그램과 실시간 작업
Type 3: 스트림 인터페이스	스트림을 통해서 통신하는 자바 프로그램과 실시간 작업
Type 4: 실시간 작업에 자바 프로그래밍 가져오기	JNI를 사용한 실시간 프로그램으로부터 자바 API를 호출함으로써 수행 (자바 네이티브 메소드 인터페이스).

[그림 1]은 JTRON의 구조도로서 ITRON을 기반으로 자바 가상 기계를 사용하여 자바의 특성을 수용할 수 있는 형태로 이루어져 있다.



[그림 1] JTRON 계층 구조도

[그림 2]는 자바 프로그램과 실시간 프로그램 사이의 커뮤니케이션 인터페이스의 계층별 관계를 나타내고 있다.

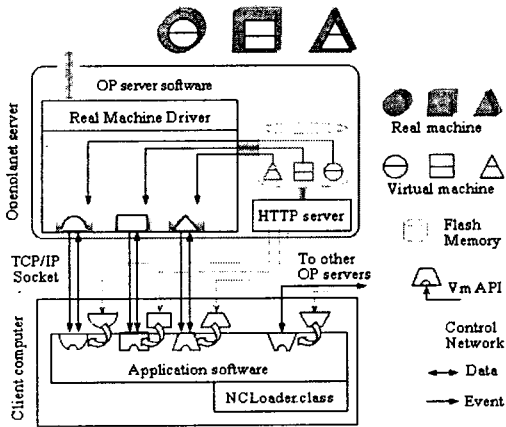


[그림 2] 자바 프로그램과 실시간 프로그램의 관계

## 2.2 OpenPLANET

OpenPLANET이란, 전기를 이용하여 동작하는 모든 기기를 인터넷 경유로 전세계 어디에서나 컨트롤할 수 있는 획기적인 '원격 감시 제어 시스템'이다. OpenPLANET은 일상생활의 안전성과 쾌적성, 편리성을 향상 시키는 것은 물론이고 여러 가지 기기의 최적 제어로 에너지와 자원을 효과적으로 활용, 21세기 고도 정보화 사회의 지구환경문제에 크게 공헌할 수 있는 기술이라 할 수 있다. [그림 3]은 OpenPLANET의 전체 구성을 나타낸 그림이다. Real machine 은 감시 제어의 대상이 되는 여러 가지 전기기기 제어계 네트워크상에서의 통신기능을 가진 원 칩 마이크로 컴퓨터를 내장함으로써 정보의 송수신 능력을 갖고 있으며, Virtual machine 은 데이터 통신에 의해 Real machine 과 동일하게 동작하는 가상 기기. 자바 언어

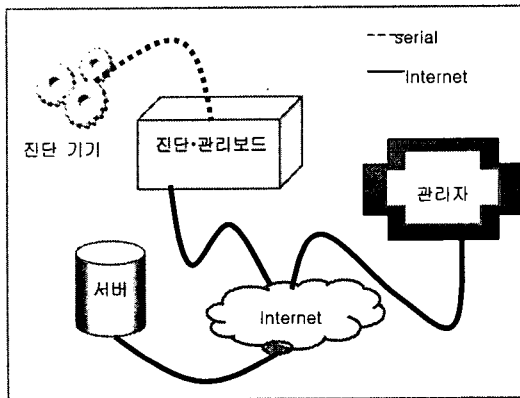
로 만들어진 소프트웨어 오브젝트로서, 인터넷 등의 정보계 네트워크상을 자유자재로 이동할 수 있다.



[그림 3] OpenPLANET 구성도

### 3. 진단·관리 시스템 설계

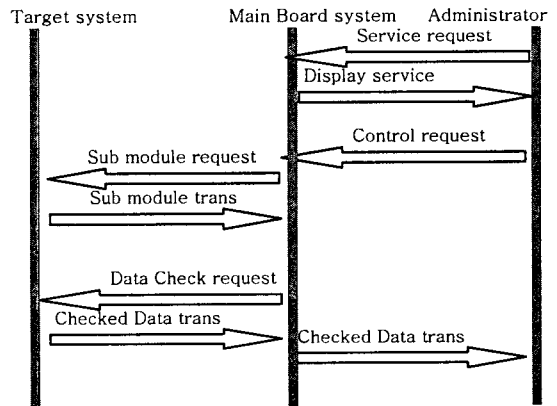
#### 3.1 진단·관리 시스템 아키텍처



[그림 4] 진단·관리 전체 시스템 구성도

[그림 4]는 본 연구에서 설계한 진단·관리 시스템의 전체 구성도이다. 진단 대상 기기는 진단·관리 보드로 시리얼 통신을 통해서 데이터를 보내주며 진단 기기와의 통신, 진단 데이터의 분석 및 처리, 관리자와의 통신 애플리케이션을 구현하기 위해 진단·관리 보드에 자바 가상기계를 탑재하여 진단·관리 시

스템을 구성한다. 진단 데이터의 분석 및 처리 과정에서는 진단 대상 기기로부터 수신된 데이터는 무결성을 검사하게 되는데, 만일 데이터에 이상이 있을 시에는 상황대처 알고리즘을 적용하여 사전 제어를 하거나 잠시 진단 기기를 정지시킨 후 관리자의 제어를 받게 된다. 진단·관리 보드와의 통신을 위한 서버와 관리자 부분은 진단·관리 보드의 IP 주소로 접속을 한 후 보드의 처리 과정이 관리자의 웹 브라우저를 통해 정보가 디스플레이 되어지며 관리자는 디스플레이된 상태를 검사하여 진단기기를 제어할 수 있도록 설계되었다.



[그림 5] 진단·관리 시스템의 실행 순서도

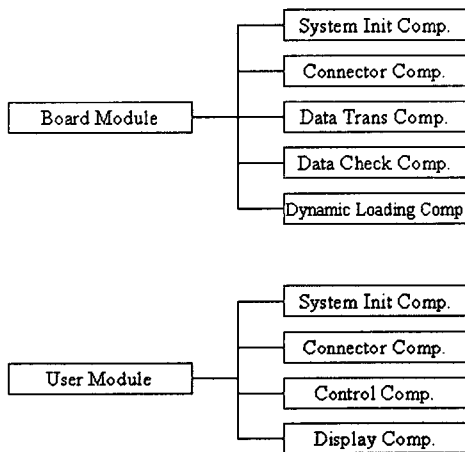
진단·관리 시스템의 구성요소는 [그림 5]와 같이 Administrator, Target System, Main Board System의 세 부분으로 구성이 된다. Main Board System 부분과 Target System 부분이 진단·관리 보드 부분에서 동작될 것이며, 관리자 부분인 Administrator가 인터넷 환경을 통해 동작될 것이다.

진단·관리 시스템의 처리과정은 먼저 Administrator가 Main Board System으로 접속하여 서비스를 요청하게 되면, Main Board System은 시스템 모니터링 및 제어에 필요한 컴포넌트들을 Administrator로 전송하게 되고 Administrator는 웹 페이지를 구축하고 Main Board System으로부터 데이터를 수신한 후 대상기기의 상태를 진단 및 모니터링 하게 된다. 모니터링 중 Administrator의 제어 요청에 따라 선택된 Main Board System을 거쳐서 Target System으로 이 요청이 전달되

고 요구하는 서버 모듈을 Main Board System으로 다운로드 하게 된다. 다운로드 된 서버 모듈은 메인 모듈의 종료 후 실행되며, 이 때부터는 선택된 Target System으로 데이터를 먼저 전송하여 진단을 끝낸 후 Administrator로 그 결과와 데이터를 전송하게 된다. 이로써 관리자는 산업기로부터 수신한 데이터와 그 데이터의 진단 결과를 확인 할 수 있게 된다.

#### 4. 구현 및 고찰

본 시스템은 인텔사의 StrongARM CPU(SA-1110), 16MB Flash Memory, 16MB SDRAM의 보드 시스템을 사용하였다. 그리고 커널 2.4버전의 리눅스 운영체제와 Java 1.1.8 스펙에 따르는 자바 플랫폼 환경을 구성하였다. 진단·관리 시스템 구현은 [그림 6]과 같이 진단·관리 보드에서 동작할 Board 모듈과 관리자를 위한 User 모듈로 크게 두 부분으로 나누었고, 각각의 모듈들은 시스템 초기화, 상호 연결, 데이터 처리, 디스플레이 기능들을 가지는 컴포넌트들로 구현되어져 있다.



[그림 6] 진단·관리 시스템 구성 모듈

#### 5. 결론

다양한 종류의 산업용 기기들을 진단·관리 하기 위한 시스템을 개발하는 과정에는 대상 기기에 대한

소프트웨어적인 이해뿐만 아니라 하드웨어 부분에 대한 이해를 요하게 되고 이는 곧 개발자의 부담으로 이어지게 된다. 본 논문에서는 자바 가상 기계를 탑재한 임베디드 시스템을 사용하여, 특정 시스템의 하드웨어 구조에 대한 이해가 없이도 시스템 개발을 용이하게 할 수 있는 플랫폼 환경을 구현 하였다. 이와 더불어 자바의 동적 다운로드 기능을 이용하여 고가의 산업 기기와 같은 새로운 기기로의 교체가 어려운 시스템을 필요 시 마다 업데이트 시킬 수 있도록 하는 진단·관리 시스템 환경을 구현해 보았다. 이러한 시스템 환경을 구축함으로써 시스템에 드는 추가적인 투자비용을 절감 하는데 상당부분 기여 할 것으로 본다.

#### [참고문헌]

- [1] Ken Sakamura and Hiroaki Takada, "ITRON4.0 Specification", TRON Association,
- [2] Yukikazu and Nakamoto, "JTRON 2.0 Specification", TRON Association, 1999.
- [3] <http://www.acunia.com/wonka/>, ACUNIA, Wonka
- [4] <http://kesl.org/>, Korea Embedded System of Linux Homepage
- [5] <http://www.tron.org/>, TRON ASSOCIATION.
- [6] <http://www.t-engine.org/>, T-ENGINE FORUM.
- [7] 이현우, 천영환, Java Programming Bible, 영진 출판사, 2001
- [8] 박영환, "임베디드 시스템 임베디드 리눅스", 사이텍 미디어, 2002