JMF 기반의 실시간 원격 공정 감시 및 제어 시스템 설계 및 구현

김 삼 룡[†] · 임 인 택[††] · 이 정 배[†††]

요 약

본 논문에서는 I/O 인터페이스 제어 보드와 ISA 인터페이스 제어 보드, 그리고 레고 기반의 프로토타이핑(Prototyping)을 통해 실시간 신발 공정 컨베이어 제어를 위한 원격 감시 및 제어 시스템을 구현하였다. 컨베이어 서버에서 실시간으로 공정을 제어하기 위하여 작업의 우선권을 사용자가 정할 수 있는 RTLinux(Real-Time Linux)를 사용하였으며, RTLinux 기반으로 디바이스 드라이버를 작성하였다. 이를 통해 Client/Server 분산 환경 시스템의 구현 및 테스트를 하며, JMF(Java Media Framework) 기반의 모니터링 시스템을 구현하여 전체 공정 상황을 확인하였다.

Design and Implementation for the Remote Real Time Process Monitoring and Control System based on the JMF

Sam-Ryong Kim[†] · In-Taek Lim[††] · Jeong-Bae Lee[†††]

ABSTRACT

In this thesis, remote conveyor control and monitoring system is designed and implemented for shoes production process based on Lego prototyping mechanism by using the I/O interface control board for the ISA interface control board. A conveyor server for real time process control is implemented based on RTLinux(Real-Time Linux) so that users decide priority of work. And device driver is implemented based on the RTLinux. Through this system, real-time Client/Server monitoring environments is implemented and tested. And the JMF(Java Media Framework) based monitoring system was implemented and the total processing status was monitored.

키워드 : 웹(Web), 분산 멀티미디어(Distributed MultiMedia), 원격 감시 및 제어(Remote Monitoring and Control), 클라이언트/서버(Client/Server), JMF(Java Media Framework)

1. 서 론

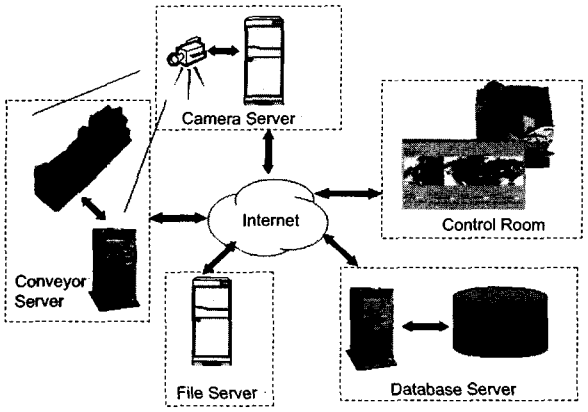
통신망 기술의 발전은 고속의 화상 처리 서비스의 제공을 가능하게 하였으며, 초고속 통신망의 등장으로 다양한 서비스를 제공하는 환경이 구축될 수 있으므로 웹 기반 원격 영상 감시 및 제어 시스템의 실현이 가능해 졌다. 디지털 비디오 기술을 이용한 원격 상황 감시 시스템으로 상황실에서 원격지의 상황을 직접 관리 할 수 있게 되었고, 제어 시스템 장애 발생 시 원격지 영상/음향 신호의 도움으로 신속하게 상황을 판단하거나, 각 장치들의 운용상태 확인, 비정상 사건상황의 화상/음향 텍스트 데이터 형태로 기록 저장하거나 필요에 따라 전송하는 요구가 확산되고 있다.

본 논문에서는 이러한 멀티미디어 제어관련 API 및 SDK를 보강한 자바 기반의 JMF[13, 14]를 사용하여 영상 스트리밍을 제어하며, JMF에서 지원하는 RTP(Realtime Transfer Protocol)[15]를 통해서 영상 정보를 실시간으로 웹을 통해 전송하고자 한다. 그리고 본 시스템은 JAVA 계열 기술인 JAVA, JSP, JDBC 등의 기술을 사용하여 제작하였다. JAVA 언어의 설계 목표 중 하나가 네트워크 상에서 플랫폼에 독립적으로 실행 가능한 응용 프로그램을 작성할 수 있게 하는 것이기 때문에 구현 방식도 기존의 구현 방식과는 다르다.

자바 언어 구현 방법으로 널리 알려진 방법은 자바 가상 기계(JVM)를 이용하여 플랫폼에 무관하게 해석적으로 실행하는 인터프리터 방식이다. 이와 같은 구현 방법은 각 플랫폼에 가상기계 인터프리터를 설치함으로써 자바 프로그램이 바이트 코드 형태로 플랫폼에 무관하게 실행될 수 있게 해준다. 또한 웹 기반 원격 감시 제어 시스템의 실시간 제어를 지원하기 위해서 컨베이어 서버는 RTLinux(Real Time Linux)[7] 시스템을 사용한다. 이러한 RTLinux 기반에서 인터페이스 카드를 통한 컨트롤 보드를 제어하기 위해서는 디바이스 드라이버[9]의 설계와 디바이스 드라이버

† 종신회원 : 경남정보대학 컴퓨터정보계열 교수
†† 정 회 원 : 부산외국어대학교 컴퓨터 · 전자공학부 교수
††† 종신회원 : 선문대학교 컴퓨터정보학부 교수
논문접수 : 2003년 7월 4일, 심사완료 : 2003년 12월 3일

를 통한 하드웨어를 쉽게 제어해 줄 수 있는 API의 설계가 필요하다. 본 논문에서 구성하고자 하는 목표 시스템은 (그림 1)에 나타나 있다.



(그림 1) 목표 시스템 구성도

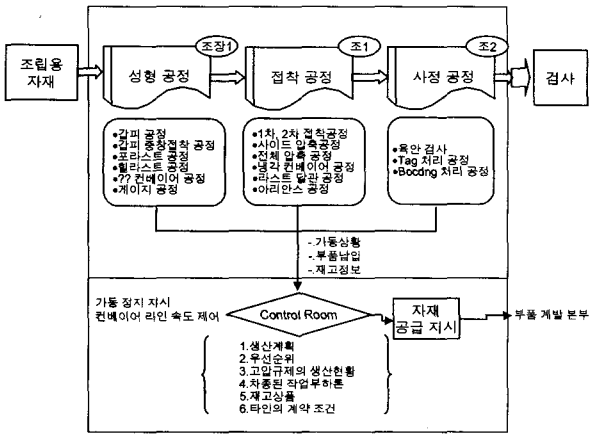
Client/Server 환경에서 초고속 정보 통신망과 분산 멀티미디어를 이용하여 컨베이어 시스템을 원격으로 제어하는 시스템의 개발을 목표로 하고 있다. 즉, 원격지에 있는 컨베이어 시스템을 원격지 현장에서 모든 공정을 통제하는 것과 같은 효과를 상황실 클라이언트 시스템을 통하여 이룰 수 있도록 한다. 이는 일대일 화상 문답 채널을 할당함으로써 상황실에서 원격 제어가 가능하도록 하게 된다. 카메라로 촬영된 영상이 카메라 서버를 통해서 웹을 통해 클라이언트가 있는 상황실로 전송 되어진다. 그리고 관리자의 요청에 따라, 파일서버에 원하는 파일 포맷으로 변화시켜서 저장 하며, 이러한 모든 이벤트들은 데이터베이스 서버에 저장이 되게 구성한다.

그러나 이러한 기술이 적용되는 공장 자동화 같은 대규모 공정의 개발에 있어서 구현과 테스트는 시뮬레이터에 그친다는 한계가 있다. 이를 위해 본 논문에서는 신발 공정 컨베이어 구성을 위해 Lego와 I/O 인터페이스 제어보드, ISA 인터페이스 제어 보드 그리고 RTLinux를 통해, 실제 모형과 비슷한 신발 공정 컨베이어 프로토타입을 구성하고, 실시간 감시 및 제어를 통해 시뮬레이터에 그치는 대규모 공정에 대해서 정확한 구현과 테스트를 할 수 있도록 제안하고자 한다.

2. 컨베이어 시스템 분석

2.1 제품의 제조 공정 분석

본 논문이 적용하고자 하는 신발 산업의 신발 제조 공정은 제조 대상이나 자재 등에 따라 달라지나 가장 일반적인 공정은 (그림 2)와 같이 성형공정, 접착공정, 사정공정 순으로 이루어진다. 성형 공정이란 신발의 갑피, 중창, 포라스트, 힐라스트 등의 성형을 담당하는 공정이며, 접착 공정은 갑피, 밑창, 등의 접착 및 냉각과 압축 및 아리안스(박음질) 공정을 총칭하는 말이다. 그리고 사정 공정은 완성 된 신발의 화학약품 처리, 본드/게이지 제거, 신발 끈 처리, 포장, 패키지 등을 담당하는 작업을 의미한다.

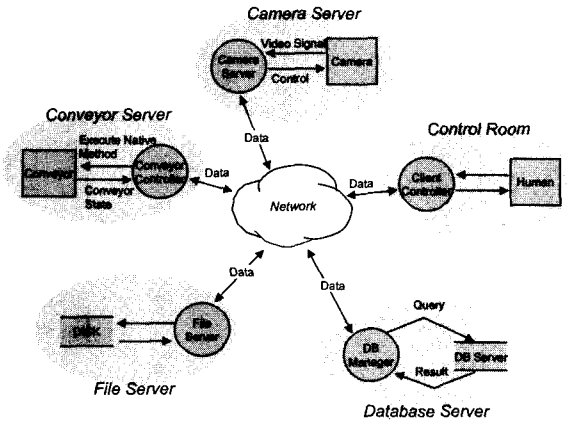


(그림 2) 신발 제조 공정

본 시스템에서는 웹을 통하여 이상의 신발 산업의 각 공정을 원격 감시하며, 다음의 세부 동작을 직접제어 또는 원격 제어하도록 한다.

- 각 공정 컨베이어 벨트의 속도 및 제한 속도 감지, 유지를 위한 제어
- 원격지 신발 접착을 위한 온도측정 및 온도유지를 위한 제어
- 각 공정별 불량품 제거
- 각 공정별 불량품 및 완제품 처리 공정 DB화

2.2 전체 시스템 구성

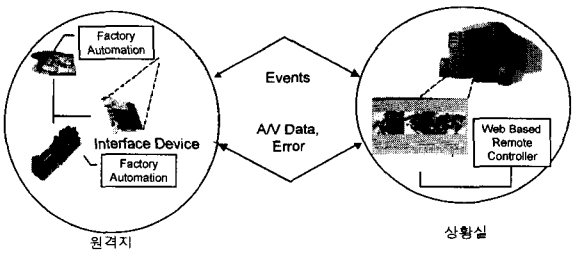


(그림 3) 시스템 구성도

본 시스템은 크게 원격지의 공정을 감시 및 제어 할 수 있는 Control Room과 현장에서 직접 공정을 제어하는 서버 부분으로 나눌 수 있다. 서버 부분은 다시 컨베이어 서버, 카메라 서버, 파일서버 그리고 데이터베이스 서버로 구

분할 수 있다. 컨베이어 서버는 공장에서 각 컨베이어 공정을 직접 제어하거나 원격지의 상황실에서 받은 제어 신호를 컨베이어 시스템에 전달하여 이를 제어하고 각 공정의 상황을 원격지의 상황실에 전달해 준다. 카메라 서버는 공장의 상황을 촬영하여 원격지의 상황실에 실시간으로 디스플레이 해주는 역할을 하며, 파일 서버는 이렇게 촬영된 영상을 저장해 두었다가 필요할 때 재생할 수 있도록 지원해 준다. 그리고 마지막으로 데이터베이스 모듈은 공정 분석을 위한 데이터들의 처리를 위한 부분이다. 이상의 5개의 시스템 구성 요소들은 원격지에서의 실시간 제어 및 감시를 위해 모두 웹으로 연결되어 있어야 하며 TCP/IP 통신과 RFP (Request for proposal) 통신 프로토콜을 사용한다.

- (그림 4)와 같이 상황실에서는 감시 영상을 통해 애플릿 그래픽 인터페이스로 제어신호(Event)를 보내고 원격지의 공장에서는 상황실로부터 전송 받은 신호를 Interface Device를 통해 컨베이어를 제어한다. 컨베이어 감시한 영상과 각 공정에서 발생하는 문제점들은 다시 상황실 영상과 기타 전달매체로 전달하여 관리자가 쉽게 상황을 판단할 수 있도록 한다.



(그림 4) 상황실과 원격지 공장과의 관계

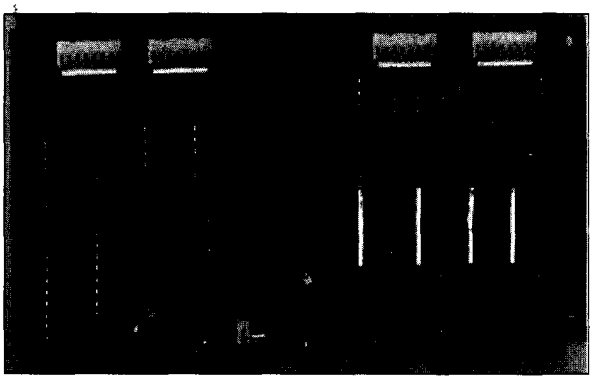
2.3 제품의 하드웨어적 구조 및 기능

- 신발 산업용 컨베이어를 위한 웹 기반 원격 제어 감시 시스템에서 사용되고 있는 컨베이어는 미국 MIT 대학과 Lego 사가 공동 연구한 레고 마인드 스톰(Lego Mind Storm) 및 I/O Interface 보드, 인터페이스 제어 카드 등으로 구성되어 있다.

- 신발 산업용 컨베이어를 위한 웹 기반 원격 제어 감시 시스템의 컨베이어 모형은 마인드 스톰의 블록과 몇 개의 센서를 이용함으로 제작되었다.

- 신발 산업용 컨베이어를 위한 웹 기반 원격 제어감시 시스템에서 사용되고 있는 레고 마인드 스톰용 센서 및 Actuator는 다음과 같다.
  - 수동센서(Passive Sensor) : 터치 센서, 온도 센서
  - 능동센서(Active Sensor) : 광 센서, 회전 센서
  - Actuator : 컨베이어에 사용되는 모터
  - 온도센서는 공장 모형의 온도 감지에 사용되는 센서이며, 본 시스템에서는 하나가 사용되고 있다.
  - 광 센서는 컨베이어를 통과하는 물체감지를 위한 센서이다. 본 시스템에서는 불량품과 완제품을 색깔에 따라 구분하고 있으며, 모두 세 개가 사용되고 있다.
  - 회전센서는 모터의 회전수를 체크하는 센서이다. 불량제거 모듈에서 사용되고 있으며 총 2개가 사용되고 있다.
  - Actuator중 본 시스템에서 사용되고 있는 것은 모터인데 이는 컨베이어의 구동, 온도조절 팬의 구동, 불량제거 모듈 등에서 총 6개가 사용되고 있다.

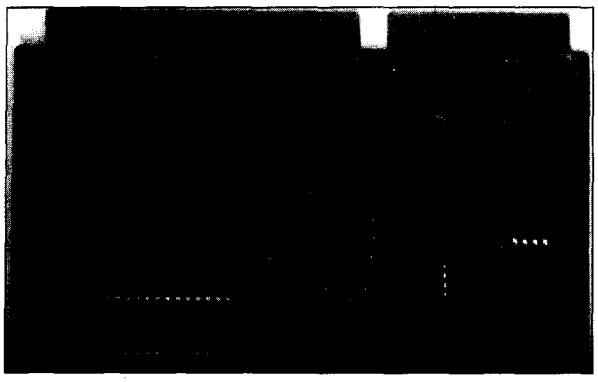
- I/O 인터페이스 제어 보드
  입/출력 장치들을 제어하는 I/O 제어 보드는 다음 같은 구조를 가지고 있다. (그림 5)가 I/O 인터페이스 제어 보드이다.



(그림 5) I/O 인터페이스 제어보드

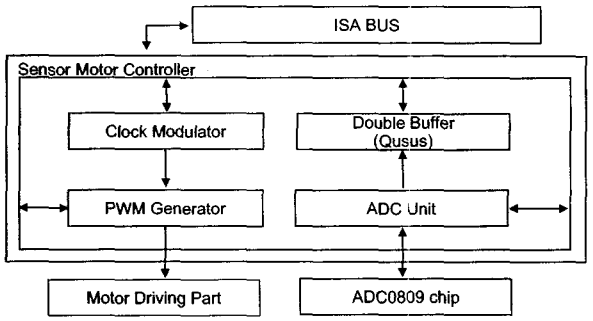
  - 입력 장치용 인터페이스 회로 : 총 8개의 센서를 동시에 연결해 사용할 수 있으며, 특히 수동/능동센서를가리지 않고 어느 채널에든 연결해 사용할 수 있다. 이 회로는 센서에 전원을 공급하고, 센서의 아날로그출력을 ISA 인터페이스 제어 보드에 있는 ADC칩에 공급해주는 일을 한다.
  - 출력 장치용 인터페이스 회로 : 총 8개의 actuator들을 동시에 연결할 수 있는 출력 회로에서는 DC모터 들에 충분한 전력을 공급하기 위해 스테핑 모터 응용에도 사용하는 L298칩을 사용한다. 이 L298을 제어 하는 신호는 ISA 인터페이스 제어 보드에서 만들어진다.

- ISA 인터페이스 제어 보드
  PC용 ISA 인터페이스 제어 보드는 프로그램의 명령에 따라 I/O 인터페이스 보드의 운영 방식을 결정하고, 현재 상태와 데이터를 수집하여 보고하는 일을 한다. 이 제어 보드는 메인보드의 ISA 슬롯에 장착되어 동작하며, ISA 아키텍처를 따라 I/O R/W(Read, Write)와 DMA, 인터럽트를 사용한다. (그림 6)은 인터페이스 제어 보드의 사진이다.



(그림 6) 인터페이스 제어보드

이런 다양한 기능을 구현하기 위해 제어용 칩으로 ALTER 사의 Flex10k FPGA 칩을 사용하였다. 이칩은 VHDL과 같은 HDL 언어로 그 기능을 구현하게 된다. (그림 7)은 제작된 VHDL 소스 코드의 기능 구성도이다.



(그림 7) Flex10k 제어 칩의 구조

(그림 7)의 중앙에 큰 테두리를 두르고 있는 부분이 flex10k 칩에 내장된 기능들이며, 외부의 ISA BUS, Motor Driving Part, ADC0809 chip은 이 칩과 연결되는 외부 회로를 나타낸다.

Flex10k 칩에는 크게 4가지의 기능을 부여했다.

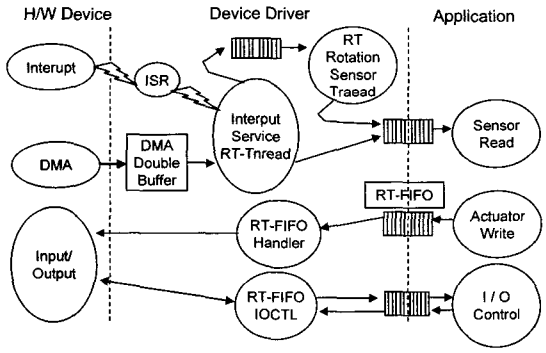
센서를 제어하는 파트, 액추에이터를 제어하는 파트, ISA I/O R/W를 담당하는 파트, 마지막으로 DMA를 통해 센서 데이터를 메인 메모리로 전송하는 파트이다.

센서를 제어하는 파트는 ADC0809 chip을 제어하는 ADC Unit이며, 액츄에이터를 제어하는 파트는 모터의 속도와 방향을 제어하기 위한 PWM 신호를 만드는 PWM Generator와 Clock Modulator이다. 이 중 PWM Generator 부분은 8개의 엑츄에이터 채널 각각에 대한 독립적인 제어가 가능하도록, 8개의 동일한 모듈 개체가 만들어져 있다. 유저 프로그램에서는 각 엑츄에이터 채널에 대해 서로 다른 PWM 펄스 주기를 설정하여 동작 방향과 속도를 제어할 수 있다. Clock Modulator는 전체 PWM Generator에 공급되는 기준 클럭의 주기를 조절하며, 8개 PWM Generator 전체의 펄스 주기에 영향을 준다.

ISA I/O R/W 명령을 받아 처리하는 부분은 Sensor Motor Controller 부분이며, DMA를 통해 8채널 센서들의 데이터를 메인 메모리로 전송하는 부분은 Double Buffer에서 담당하고 있다. 특히 이 Double Buffer에서는 128바이트의 버퍼 두개가 Flex10k 칩 안에 내장되어서 ADC칩에서 넘어오는 데이터를 보관하여 두었다가 DMA로 한꺼번에 전송하도록 구성되어 있다. 또한, 소프트웨어 측의 필요에 따라 DMA를 거치지 않고 직접적으로 ADC칩에서 받은 가장 최근의 센서 값을 받을 수도 있도록 기능을 추가하였다.

- 디바이스 드라이버와 API 제작

디바이스 드라이버는 유저 프로그램의 요청을 받아 제어 보드에 명령을 내리거나, ISA 인터페이스 제어 보드에서정보를 받아오는 일을 하는 시스템 프로그램이다. 일반 Linux에서의 디바이스 드라이버는 유저 프로그램의 요청을 받아 서비스를 하는 구조로 되어 있으며, 이러한 디바이스 드라이버의 작동 타이밍은 Linux 커널에서 임의로 정하게 되어 있다[8]. 이는 제한 시간 내에 정해진 작업을 처리해야 하는 실시간 시스템 특성에는 그리 적합한 구조가 아니다. 예를 들어 Bottom Half와 같은 부분에서는 커널의 원인으로 제한 시간을 초과하는 상황이 생길 수도 있기 때문이다. 이에 비해 RTLinux의 경우는 작업의 우선권을 개발자가 설정할 수 있으므로, RTLinux기반으로 제작된 디바이스 드라이버는 일반 Linux기반의 디바이스 드라이버보다 실시간 시스템에 더 적합하다. (그림 8)는 실제 제작된 RTLinux용 디바이스 드라이버의 구조도이다.



(그림 8) 디바이스 드라이버의 기능적 구조도

제작된 디바이스 드라이버는 크게 4가지 기능으로 구성되어 있다. 첫째는 DAM를 통해 전달되어 오는 센서 측정값을 유저 프로그램으로 넘겨주는 부분이다. 여기서는 과다한 DAM 데이터 량을 조절해 I/O 트래픽을 줄여주는 필터를 삽입하여 효율성을 향상 시켰다. 필요에 따라 피크 노이즈를 제거하는 메이안 필터도 사용 가능하다.

두 번째는 액츄에이터로 전달되는 데이터를 처리하는 부분이다. 여기서는 유저 프로그램의 명령을 받아 제어 보드에 넘겨주는 일을 한다. 디바이스 드라이버의 명령을 받은 제어 보드는 해당 엑츄에이터 채널의 PWM 펄스의 주기를 조절한다.

세 번째는 입/출력 데이터 외에 Flex 칩에 전달되는 제어 명령을 주고받는 부분이다. 이 작업은 유저 프로그램의 요청에 바로 결과를 알려주게 되며, 이러한 양방향성을 이루기 위해 두개의 RT-FIFO를 사용한다. 필요에 따라 유저 프로그램에서는 하나의 FIFO를 이용하는 것으로 보이도록 할 수도 있다.

네 번째는 응용 프로그램에서의 필요에 따라 삽입된 것인데, 바로 회전 센서의 회전 방향과 회전수를 측정하기 위한 루틴이다. 이것은 고속의 회전 환경에서 정확하게 회전 방향과 회전수를 계산하기 위해 유저 프로그램의 필요에 따라 추가되어진 기능이다.

제작된 디바이스 드라이버에서는 이러한 각 기능 단위의 프로그램 인터페이스마다 RT-FIFO를 각각 사용하도록 구성하였으며, 총 5개의 RT-FIFO를 사용하고 있다.

디바이스 드라이버 제작 후에는 이를 쉽게 사용하기 위해 응용 프로그램을 위한 API를 제작하였다. 기본적인 입출력을 고려한 API들이 제공되며, 응용 사례에 따라 얼마든지 기능이 추가되어 질 수 있을 것이다.

3. 시스템의 설계

3.1 JMF를 이용한 영상 정보의 캡쳐, 저장 및 RTP를 이용한 실시간 송/수신

JMF라는 것은 기존의 자바 프로그래밍에서의 가장 취약했던 멀티미디어 제어관련 API 및 SDK를 보강한 것으로서, 윈도우의 VIdeo for windows 프로그래밍 SDK와 마찬가지로, 비디오 및 오디오의 캡쳐, 저장, 전송, 스트리밍 기술 등에 이용되는 기술이다. Java Media와 Communication APIs는 오디오, 비디오, 애니메이션, 3D그래픽과 같은 발전된 멀티미디어 타입 창출과 플래이백(Playback)을 지원하는 인터페이스로서 설정되어 있다.

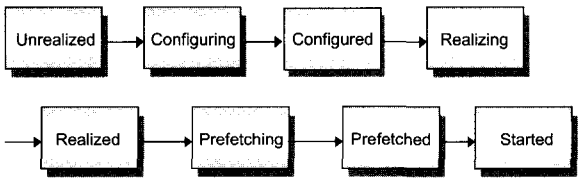
JMF에서는 MPEG, AVI, MOV등의 비디오 기술과 더불어 H.261 H.263등의 영상회의 코덱 표준도 지원하고, G.721 G.723 등의 오디오 코덱도 더불어 제공하고 있다. 또한 RTP (Realtime Transfer Protocol)을 지원함으로서 실시간으로 동영상과 음성을 전송하고, 이러한 기능을 통해서 인터넷 멀티 채팅, 화상회의, 원격 감시등에도 그 이용분야가 상당히 높다.

3.2 영상 정보의 캡쳐(Capture)

영상을 캡쳐하기 위해서 컴퓨터에 설치된 Capture Device의 정보를 얻고, 그 Capture Device에 어떤포맷이 있는지 조사하여 사용가능한 포맷을 정한다. 본 논문에서는 캡쳐 장치로 USB Camera를 사용하였다. 실제로 하드웨어에 장착되어있는 캡쳐 디바이스에 대한 정보를 관리하는 관리자는 CaptureDeviceMan-ager이며, 장치에서 사용할 수 있는 포맷이 어떤 것이 있는지 알아보기 위해서 CaptureDeviceInfo 클래스를 이용하였다. 이렇게 하드웨어 적인 캡쳐 장치를 인식하고, 그 장치에 대한 드라이버 정보를 얻어오고, 포맷을 구분하고, 원하는 포맷을 찾아서 그 장치의 이름을 얻어오고 나서, 그 장치로부터 실제 데이터 소스인 영상을 캡쳐하게 하였다.

3.3 영상 정보의 저장

캡쳐된 영상 정보를 저장하기 위해서는 내부적으로 데이터를 프로세싱하고 원하는 데이터 포맷으로 부호화를 한 후에 미디어 기록장치에 기록을 하게 된다. 여기서 재생기(Player)는 프로그램이 가능하기 때문에 디코딩과 랜더링 처리를 가능하게 해 준다. 또한 캡쳐 프로세서로 이용됨으로서 캡쳐된 멀티미디어 데이터에 대한 부호화 및 멀티플랙싱 기능을 수행 할 수 있다. (그림 9)는 프로세스 상태도를 나타낸 것이다.



(그림 9) 재생기 상태도

재생기가 Unrealized의 상태에서 configure 메쏘드를 호출하면 재생기는 Configuring 단계로 들어가게 된다. 이러한 Configuring 단계는, 재생기가 바로 DataSource에 연결을 수행하고, 입력된 데이타 스트림을 디멀티플랙싱 시키며, 입력 데이타의 데이타 포맷에 관한 정보를 얻는 단계다. 재생기가 DataSource를 찾는데 성공했고, 데이타 포맷에 관한 정보획득이 성공하면, 재생기는 Configured 상태로 이동하게 되고, 재생기가 이 상태에 도달하게 되면 configureComplete Event가 발생한다. 그러면 재생기가 Realize 메쏘드를 호출하고 재생기는 Realized 상태로 이동하게 되며, 이때부터 재생기는 완전히 생성이 성공된 단계라고 할 수 있다.

재생기로부터 출력되는 미디어 데이타를 읽고 파일로 저장하기 위해서 DataSink를 이용한다. 저장 과정을 살펴보면, 먼저 API의 getDataOutput 메쏘드를 호출하여 재생기로부터 출력 DataSource를 얻고, Manager.createDataSink 메쏘드를 호출해서 file writer DataSink를 생성한다. 이것을 DataSource와 MediaLocator에 넘겨 주어서 저장하고자 하는 파일의 위치를 지정해 주고, 파일을 열기 위해서 open 메쏘드를 호출한다. 그리고 DataSink에 대해서 start를 호출하여 데이타 기록을 시작한다.

3.4 RTP를 이용한 영상 정보의 실시간 송/수신

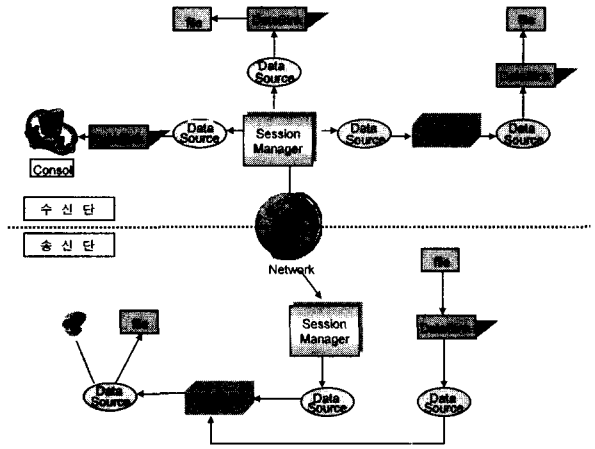
먼저 RTP에 대해서 살펴보면, 일반적인 엄격한 타이밍 요구조건을 갖는 TCP기반의 HTTP, FTP와는 달리, UDP 프로토콜인 RTP는 주로 미디어 스트림에 대한 전송을 목

적으로 설계되었으며, 모든 RTP 버퍼들은 타임스탬프(timestamp) 를 가지고 있어서, 타임스탬프와 실제 전역적인 동기화된 클럭(예를 들어서 JMF 재생기의 TimeBase)사이의 매핑(mapping)을 수행한다. 타임스탬프의 역할은 다양한 데이터 소스로부터 제공되는 미디어들을 통합하는 기능을 지닌다.

전송되는 각각의 패킷은 순서 번호(Sequence Number)와 함께타임스탬프를 갖게 되는데, 순서 번호r은 각 패킷마다 서로 다른 고유의 번호로서, 수신단에서 패킷 손실에 대한 검출과 복구에 이용된다. 타임스탬프는 동일한 번호를 지님으로서 특정한 동일시간에 함께 복호화 되야 함을 의미한다.

RTP역시 UDP 기반의 프로토콜로서 수신단에서는 수신되어지는 데이타 패킷이 일정하게 알맞은 순서로 전송되었는지 알수 없다. 그러므로 수신단에서는 패킷의 순서 번호를 이용하여 패킷을 재 정렬해야한다

네트워크를 통해서 RTP 데이터의 송신과 수신의 과정은 아래 (그림 10)에서와 같이 이루어진다.



(그림 10) RTP 데이터 송신과 수신

먼저 수신측을 살펴보면, 네트워크를 통해 전달되는 RTP 데이터는 일단 Session Manager에 의해 세션별로 분리가 되어서 각각의 별도의 데이타 소스들로 분리가 된다. 만약 수신단에 저장을 하기전에 수신단에서 알맞은 프로세싱을 하고 그 후에 저장을 하고자 한다면 데이타 소스를 프로세서로 넘기고 프로세서의 출력을 다시 데이타 소스로 만들고, 그것을 데이타 싱크로 넘긴후 데이타 싱크에서 저장을 하게된다. 또 수신단에서 수신받은 데이타를 플레이 하고자 하는 경우에는 Session Manager에서 나오는 데이타 소스를 Player에게 넘겨주면 영상을 화면에서 볼 수 있다.

송신측은, 데이터 송신을 위해 획득한 데이터들은 DataSource로 보내어 지고, 프로세싱을 위해서 Player로 들어간다. Player의 출력물로 다시 DataSource 형태로 출력물이 나오면 그것을 Session Manager를 통해 네트워크로 전송하거나, 혹은 DataSink를 이용해서 하드디스크에 저장을 하도록 구성한다.

3.5 컨베이어 서버 모듈 설계

컨베이어 서버 모듈의 구동하기 위해서 RT(RealTime) 리눅스를 사용한다.

원격지 공장의 컨베이어에서 일어나고 있는 모든 현상(온도변화, 컨베이어의 속도변화 등)은 감시부를 통해 클라이언트에 전송(Broadcasting)되도록 한다. 클라이언트에서 보내온 컨트롤 시그널은 제어부를 통해 각 공정을 제어한다. 온도 변화에 따른 컨베이어 공정내의 자체 이벤트 처리와 같이 사용자를 거치지 않아도 상관없는 부분에 대한 제어는 모듈 내에서 자동적으로 수행되도록 설계한다.

3.6 카메라 서버 모듈 설계

카메라 서버는 카메라로부터 입력받은 영상을 원격지의 관리자에게 전송하거나 파일서버에 전송하는 역할을 한다. 영상의 전송은 카메라에서는 USB포트를 통해서, 컨베이어 서버와는 TCP/IP 소켓 프로그램으로 연결한다. IP 정보와 함께 영상화면 요청을 컨베이어 서버로부터 받은 카메라 서버는 입력된 영상을 영상처리 모듈을 통하여 사용자에게 RTP 패킷으로 전송을 한다. 이러한 상황은 모두 인터넷에서 실시간으로 이루어지므로 웹에서 발생되는 트래픽을 제외하면 지연이 거의 발생하지 않는다.

4. 시스템의 구현 및 테스트

4.1 Lego RIS을 이용한 신발 공장 모형 제작

(그림 11)은 이러한 구조도에 따라 Lego를 이용해 실제 구현한 신발 공장의 모형이다.

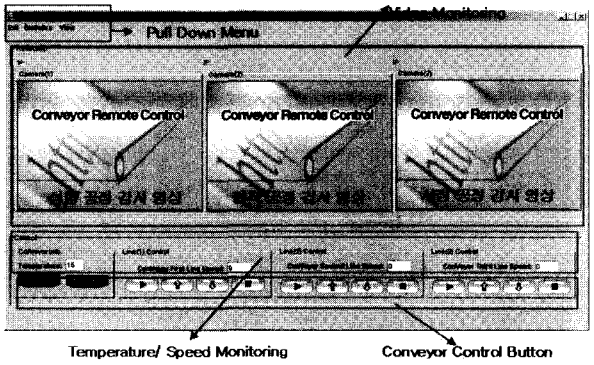
(그림 11) Lego로 제작된 신발 공장 모형

4.2 서버 모듈들의 구현

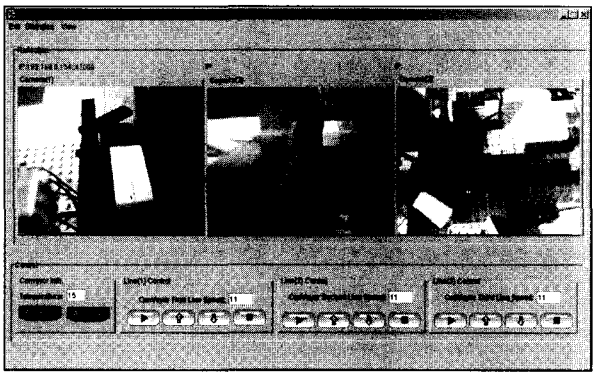
4개의 서버의 모듈들은 분산 환경 지원을 위해 각각 서로 다른 컴퓨터에서 실행 시킨다. 각 각의 서버 모듈은 플랫폼의 독립성을 위해 JAVA 언어로 제작되어 있으나, 컨베이어 서버 모듈은 JAVA와 C언어로 구성되어 있다. 이러한 JAVA와 C의 연동을 위해 JNI(Java Native Interface)를 사용한다.

4.3 클라이언트 사용자 모듈의 구현

클라이언트 사용자 모듈은 웹으로 접근할 수 있는데, 사용자 인증을 거친 홈페이지와 연동되어 있다. 클라이언트 사용자 모듈에서는 (그림 12)와 같은 원격지 공장 감시 및 제어를 위한 상황실 Main GUI와 (그림 13)과 같은 공정 분석 GUI 등이 제공된다.



(그림 12) 상황실 Main GUI



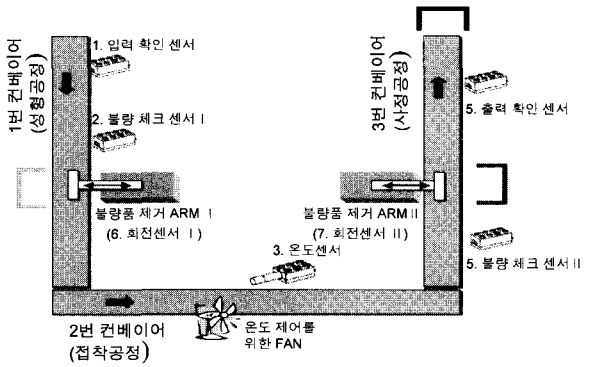
(그림 13) 공정 분석 GUI

이 외에도 사용자 클라이언트 모듈에서는 디렉토리 검색 서비스를 이용한 영상 재생 프로그램, 영상 전송 오류 시 그래프를 이용한 원격 감시 프로그램 등이 구현되어 있다.

4.4 시스템의 동작 시험

컨베이어 모형은 신발 제조의 세 가지 공정인 성형, 접착, 사정 공정을 각각 하나의 컨베이어 벨트에 축소해 놓았다. 모형에 사용되는 센서는 4개의 빛 센서와 2개의 회전 센서 그리고 하나의 온도 센서로 이루어져 있으며 동작을 담당하는 Actuator는 모두 모터로 총 7개가 사용되고 있다.

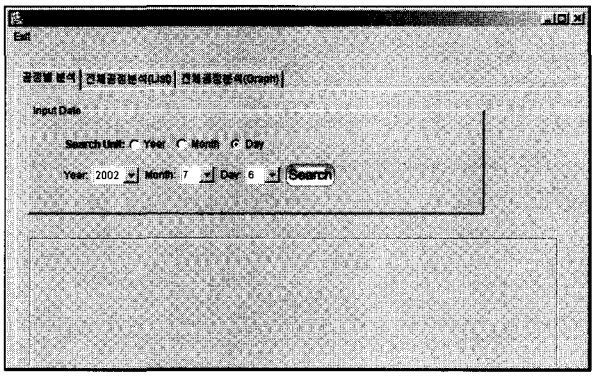
성형공정을 담당하는 1번 컨베이어로 Item이 들어온다. 입력확인 센서에서 총 투입 Item의 개수를 파악하고 불량 체크 센서 1을 통해 성형공정에서 발생하는 불량품을 확인한 후 불량품 제거 ARM I으로 불량품을 제거한다. 1번 컨베이어를 통과한 아이템은 접착공정을 담당하는 2번 컨베이어로 이동하게 된다. 접착공정에서는 온도를 체크한다. 온도센서를 통해 온도가 너무 많이 올라가게 되면 온도 제어를 위한 FAN이 반응 하게 된다. 마지막으로 3번 컨베이어인 3번 사정공정에서 불량 체크 센서 2를 통해 불량품을 다시 한번 걸러내고 출력 확인 센서를 통해 완제품의 개수를 파악하는 것이다. Lego 모형에서는 빛 센서를 이용하여 Item의 컬러로 불량체크를 하게 했다. (그림 14)는 신발 산업 각 공정에 대한 모형 구조도이다.



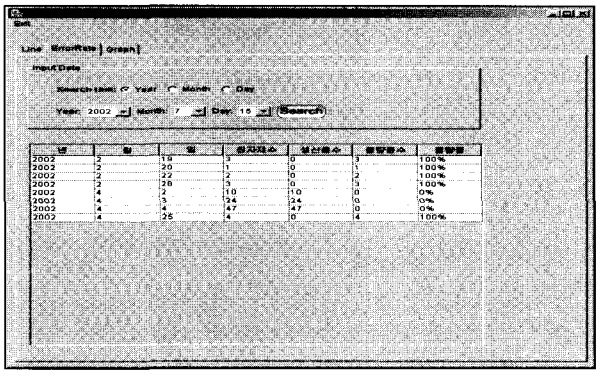
(그림 14) 신발 산업 공정에 대한 모형 구조도

다음은 각 공정을 테스트 및 분석 화면이다.

- 공정별 분석 : (그림 15)는 성형, 접착, 사정 공정별 불량품 통계를 시간대 일별, 월별로 제공한다.



(그림 15) 공정 분석 툴의 공정별 분석 실행

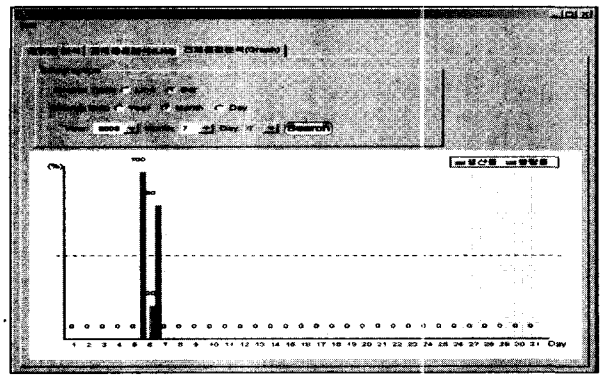


(그림 16) 전체 공정 분석

- 전체 공정 : (그림 16)은 원격지 공장 전체 공정의 완제

품 및 불량품 개수, 불량품 발생률 등을 테이블 형태로 제공한다.

- 전체 공정 분석 그래프 : (그림 17)은 원격지 공장 전체 공정의 완제품, 불량품 개수 및 불량품 발생률 등을 막대와 선, 그래프로 제공한다.



(그림 17) 전체 공정 분석(Graph)

5. 결 론

본 논문에서는 제시한 Lego, I/O 및 제어보드를 이용하면 신발 공정과 같은 대규모 실시간 시스템, 원자로, 고온/고압 산업현장 등의 극한 작업 환경에 무인 실시간 원격 제어 및 감시 기술 모두 구현해 볼 수가 있다. 또한 Client/Server 분산 환경의 웹으로 실시간 처리가 가능하기 때문에 인터넷만 연결이 된다면 지구상 어느 곳이라도 감시 및 제어가 가능하다.

그리고 공정 제어 분야에서는 공장 운영에 상당한 비용 절감을 기대해 볼 수 있고, 생산성 향상을 꾀할 수 있을 것으로 기대 된다. 그리고 앞으로 크게 확산될 것으로 예상되는 인터넷 정보 가전 및 자동 공정 제어감시 시스템의 필요성을 더욱 더 크게 만들 것이여, 온라인화 되어 있는 의료 및 가정 컴퓨터의 경우까지도 이러한 시스템을 필요로 하는 환경이 조성될 것이다. 또한 원격 제어 및 감시 기술은 국내 전자·정보통신 산업의 발달을 더욱 가속화시킬 것이므로, 고도 정보화 사회의 실현에 일익을 담당할 것이다.

참 고 문 헌

[1] http://fsmlabs.com/.

[2] http://www.lego.com/.

[3] http://mindstorms.lego.com/.

[4] http://java.sun.com/products/java-media/jmf/2.1.1/solutions/.

[5] http://kldp.org/KoreaDoc/.

[6] http://www.javastudy.co.kr/docs/jhan/javaadvance/jniexamp.html.

[7] http://www.rtlinux.co.kr

[8] 이현우, 천영환, Java Programming Bible for JDK 1.3, 영진.com, May, 2001.

[9] Alessandro Rubini. *LINUX Device Drivers*, O'Reilly, 1998.

[10] Hassan Gomaa. Software Design Methods for Concurrent and Real-Time Systems, Addison-Wesley, 1996.

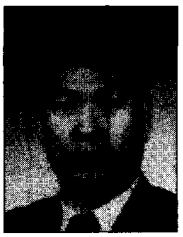
[11] 정기훈, 김도훈, 박성호, 강순주, "임베디드 실시간 시스템 개발 교육과정", 정보처리학회지, 제9권 제1호, Jan., 2002.

[12] 이정배 외, "신발산업용 컨베이어를 위한 웹기반 원격 제어 감시 시스템 개발 기술개발 및 사업화 결과 보고서", 한국기술평가원, July, 2002.

[13] http:/myhome.naver.com/kingseft/gallery JMF.html.

[14] Gordon, Rob/Talley, Stephen, Essential JMF, Prentice Hall Ptr, Jan., 1999.

[15] Perkins, RTP : Audio and Video for the Internet, Addison Wesley.



김 삼 룡

e-mail : ks889r@kit.ac.kr

1974년 동아대학교 공과대학 전자공학과 공학사

1986년 동아대학교 대학원 전자공학과 공학석사

2000년~현재 부산외국어대학교 컴퓨터·전자공학부 박사과정

1981년~현재 경남정보대학 컴퓨터정보계열 교수

관심분야 : 인터넷 프로그램, Mobile IP, MIS, e-Learning

임 인 택

e-mail : itlim@pufs.ac.kr

1984년 울산대학교 전자계산학과 공학사

1986년 서울대학교 대학원 계산통계학과 이학석사

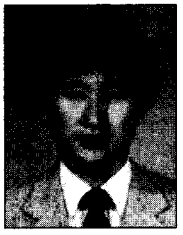
1998년 울산대학교 대학원 컴퓨터공학과 공학박사

1986년~1993년 삼성전자(주) 특수연구소 선임연구원

1993년~1998년 동부산대학 전자계산과 조교수

1998년~현재 부산외국어대학교 컴퓨터·전자공학부 조교수

관심분야 : 무선 ATM망, Mobile IP, MAC프로토콜



이 정 배

e-mail : jblee@sunmoon.ac.kr

1981년 경북대학교 전자공학과 전자계산전공 공학사

1983년 경북대학교 대학원 전자공학과 전자계산전공 공학석사

1995년 한양대학교 대학원 전자공학과 정보통신전공 공학박사

1982년~1991년 한국전자통신연구원 선임연구원

1991년~2002년 부산외국어대학교 컴퓨터·전자공학부 부교수

2002년~현재 선문대학교 컴퓨터정보학부 부교수

관심분야 : 실시간 시스템, 임베디드 시스템, 실시간 통신 프로토콜