

인터넷 기반의 온실 환경 제어 시스템에 관한 연구

김 대 업[†] · 박 흥 복^{††}

요 약

현재 국내에 보급되어 있는 시설 원예를 위한 온실 환경 제어 시스템은 노동 집약적 시설에 타이머를 이용하여 각 장치를 제어하는 on/off 식 제어반으로 구성되어 있다. 이러한 시스템은 하드웨어적인 결함, 야간이나 원격지에 있을 경우에는 오류 발생 등에 대한 온실의 상태 파악이 어렵다. 따라서, 본 논문에서는 기존의 on/off 식 제어 방식을 현장에서는 액정 패널로 제어하도록 하였고, 온실의 각 장치를 제어하는 클라이언트 컴퓨터를 통하여 인터넷상에서 원격지의 데이터 획득과 온실의 상황을 모니터링할 수 있는 시스템을 개발하였다. 또한 각 지역별 온실과도 통신이 가능하며 오류 발생시에도 실시간으로 대처하는 경보 메시지 전송, 그리고 효율적인 온실 환경 제어 및 작물 재배 정보들을 관리할 수 있는 Web 서버를 구축하였다.

A Study on System for Environmental Control in the Greenhouse using Internet

Dae-Up Kim[†] · Hung-Bog Park^{††}

ABSTRACT

Recently, Environmental Control Devices of Green House for Protected Cultivation is composed of on/off-type Controller to control each device using timer in equipment of intensive Labor. In case of system, it is difficult to grasp condition of the greenhouse about errors when this system has defect of a hardware, and it operates in the remote place or at night.

In this paper, we developed a system that capable of replacing the existing control method of on/off with display panel in the greenhouse, monitoring data acquisition and status in the greenhouse using client computer based on internet. Also this system can communicate with each local greenhouse, and send alarm message about error. And we constructed web server to manage efficient informations for environmental control and corps cultivation in the greenhouse.

키워드 : 인터넷(Internet), 모니터링(Monitoring), 환경제어(Environmental Control)

1. 서 론

ADSL, ISDN 등과 같은 초고속 인터넷 접속 서비스가 제공됨에 따라 일반 가정에서 인터넷을 이용하여 영화나 음악을 감상하는 것이 가능하게 되었다. 이미 인터넷은 '정보의 바다'라고 부를 정도로 엄청난 양의 정보가 존재하게 되었으며, 그 이용자 수도 수천만 명에 이른다[11, 12]. 또한 인터넷의 발전과 더불어 인터넷을 사용하여 공장이나 작업장의 제어 상황에 대한 모니터링이 증가되는 추세이다. 이러한 추세에 따라 우리 나라의 시설 원예 산업은 노동 인구의 감소와 노동 비용이 증가함에 따라 재래식 온실에서부터 점차 자동화된 온실의 시설 변화가 요구되고 있으며 인터넷상에서도 온실의 상황을 관리할 수 있는 새로운 참

단 시설이 국내에 설치되고 있으나 거의 외국에서 모든 자재, 소프트웨어, 양액 제어 시스템까지 수입하여 사용하고 있는 실정이라서 수입 대체 산업을 위한 기술이 필요한 실정이다. 농업 인력 부족과 기술 미비를 극복하고 농업 생산품의 수량과 질을 높이기 위해 작물 재배를 위한 시설 자동화 기술이 국내에서도 지속적으로 개발되고 있다. 그러나 관련 정부 기관 및 연구소에서 시설 원예 산업을 육성하여 왔지만, 현재 국제화라는 큰 흐름 앞에서 국제 경쟁력을 키울 수 있는 장 단기 계획이 미흡할 뿐만 아니라 노동 생산 인력조차도 점차적으로 감소되고 있는 실정이다. 더욱이 농촌 지역의 생산 인력 고령화됨에 따라 공간 및 시간 제약적인 생산 환경에 적절한 대응을 하지 못하고 있다[1]. 기존 시스템의 모니터링 및 농산물 관련 전자 상거래의 경우 사용자가 일일이 확인되어야만 하고 장소의 한계성을 극복하지 못하고 있다. 모니터링의 능동적 효과 및 활용 여부는 관련 산업의 시장 활성화를 위한 많은 잠재적 가능성을 가

* 본 논문은 부경대학교 발전기금지원비에 의해 수행되었음
† 준 회원 : 부경대학교 대학원 전자계산학과
†† 정 회원 : 부경대학교 전자계산학과 교수
논문접수 : 2001년 4월 17일, 심사완료 : 2001년 6월 25일

지고 있다. 현재 시설 면적의 대부분을 차지하고 있는 비닐 온실의 경우, on/off식 제어반은 온실 설치 비용과 사용자가 항상 상주해야 하는 번거로움이 있으며, 기계 오작동으로 인한 위험이 존재한다. 따라서 대부분의 사용자는 야간이나 원격지에 있을 때에 아주 불편함을 느끼며, 기존 시스템의 각 자원에 대한 기계적인 부품의 결합으로 농가의 작물에 치명적일 수가 있다. 또한, 관련 시설 생산 시장의 크기가 협소하고 관련 연구도 아직은 많이 미흡하다[2,3]. 또한, on/off 식의 제어반을 사용함으로써 발생하는 하드웨어적인 오류, 야간 또는 원격지에서 온실 상황을 알 수 없는 문제점이 있다. 이러한 문제를 해결하기 위해서 기존의 연구[5,6]에서는 시뮬레이션 환경에서 작물 생산 과정에 필요한 정보를 수집, 감시, 저장, 제어를 수행하는 효율적인 생산 시스템을 설계하고 시뮬레이션하였다. 기존의 논문[6]에서 구현된 온실 관리 시스템은 연동 온실의 확장성과 지역 온실과의 통신, 원격지 모니터링에 대한 모듈들이 하나의 Application으로 통합 설계되어 시스템이 비효율적으로 설계되어 있었다. 또한, 하나의 Java Application으로 시스템을 구성하였기 때문에 인터넷으로 접속하려면 서버의 자원의 낭비가 심했다.

따라서, 본 논문에서는 기존의 논문[5]에서의 오류 처리에 대한 기능과 구현된 재배 작물 DB 시스템 구성 및 처리의 기능을 향상시키고, 기존의 연구[5,6]에서 설계 및 구현된 방법을 인터넷상에서 온실 환경 제어 모니터링을 위한 소프트웨어를 개발하였다. 특히, 시뮬레이션 수행시 고려되지 못한 실제 자원의 오류를 보완하였고, 시스템의 각 제어 자원의 동작 수행을 감시하는 모니터링 모듈의 실시간 자원 모니터링을 최소한의 자원으로 운영하도록 하였으며, 인터넷을 이용한 원격지의 지역 온실 모니터링을 각 모듈별로 설계 및 구현하였다. 각 모듈별로 독립적인 스레드(thread)를 할당하여 서버의 부하 및 시스템의 성능을 향상시켰다. 또한 서버에 독립적으로 주문·판매 시스템을 구현하였고 운용의 편의성을 위하여 계층적인 GUI로 구성하였다.

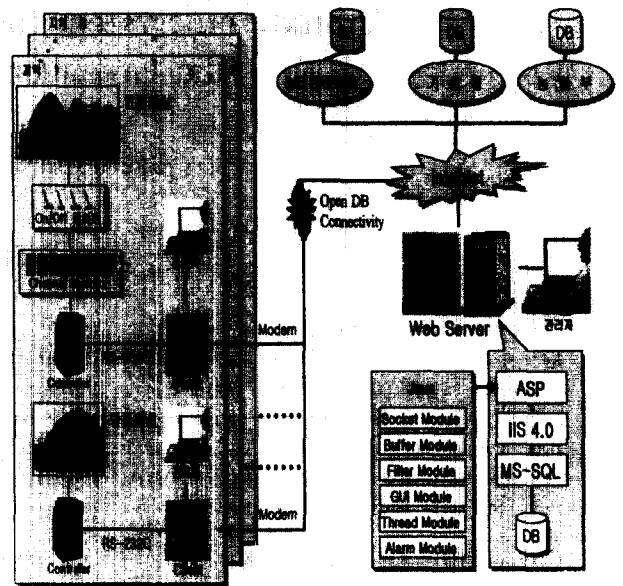
본 논문의 구성은 2장에서 시스템 설계 방법, 3장에서는 구현 및 결과 분석, 마지막으로 4장에서는 이 논문의 결론을 서술한다.

2. 시스템 설계 방법

2.1 시스템 구성도

개발된 시스템 구성도는 시스템 환경 설정 레지스트리(Registry) 파일들과 연동 온실 제어 애플리케이션, 실제 온실 자원을 직접 제어하기 위한 하드웨어와 인터넷을 이용한 모니터링, 주문·판매 시스템, 정보 시스템, 그리고 지역 온실과의 통신으로 구성된다. 클라이언트 시스템은 PC로 온실 자원을 제어하는 Controller, 기존의 해당 on/off 제어

반을 선택할 수 있는 Channel Selector, 정보 시스템, 물리적 온실 자원 등으로 구성된다. 서버는 웹 모니터링 위한 Java와 주문·판매 시스템의 인터페이스 기능을 위한 Active Server Page 모듈, 지역별 온실의 환경 정보 데이터를 서버의 파일 레지스트리에 저장하기 위한 소켓(socket) 모듈, 서버 레지스트리에 저장되어 있는 환경 정보를 저장하기 위한 Buffer 모듈, 저장되어 있는 자료를 처리하기 위한 Filter 모듈, 서버 측의 GUI 모듈, 이벤트에 따른 스레드 처리 모듈, 그리고 클라이언트와의 통신을 위한 네트워크로 구성된다. 지역 온실과 서버와는 모뎀으로 연결되어 있으며 하드웨어 컨트롤러와 컴퓨터와의 통신을 위해서는 RS-232C를 사용한다. 이 시스템의 구성도는 (그림 1)과 같다.



(그림 1) 시스템 구성도

기존의 논문[5]에서 구현된 클라이언트 제어 시스템을 현장에 적용했을 때 하드웨어 자원의 특성으로 인한 일부 부적절한 동작의 판단을 수행하였다. 본 논문에서는 시스템의 처리 동작을 판단하는 Automation Engine과 하드웨어적인 입/출력 처리를 위한 I/O Interface 모듈을 객체 지향적으로 분리·설계하였다. 따라서, 각 모듈의 기능을 전문화하며, 독립된 객체가 자신의 기능적 동작을 병행적으로 수행함으로써 시스템의 동작 판단의 적절성과 수행 속도를 높일 수 있었다. 실제 적용된 모니터링 모듈은 각 창에 대해 광센서와 동작 수행 제한 시간을 이용하여 수행 완료 여부를 점검함으로써 시스템 모니터링 제어 자원의 정확성과 비용의 효율성을 고려하였다. 기존의 논문[6]에서는 인터넷상에서의 모니터링을 하나의 Application으로 구성하여 서버 시스템의 부하가 많았으며 연동 온실에 대한 모니터링은 지원하지 않았다. 그러나 본 시스템은 각 이벤트에 대한 모듈별

로 설계 및 구현되었으며 이벤트가 종료될 때는 해당 모듈도 종료되도록 설계하였다. 연동 온실의 확장을 위하여 클라이언트와 서버와의 약속된 Packet Frame을 정의하여 연동 온실의 환경 정보를 서버에서 모니터링할 수가 있다. 모니터링 정보를 위해 우선 클라이언트 시스템에서 생성된 지역 온실의 모니터링 정보를 모델을 통해 서버 소켓으로 데이터 프레임을 송신하며, 서버는 송신된 프레임 정보를 자바의 native method와 각 모듈들을 사용하여 데이터 필터를 수행한다. 이러한 필터 과정을 통해 서버는 클라이언트의 모니터링 정보를 텍스트 형식으로 변환하여 서버의 래지스트리에 등록한다. 또한 서버를 통해 지역별 온실과의 정보 교류도 가능하다. 그리고 오류 발생시 클라이언트는 사용자 핸드폰에 연결된 SMS 게이트웨이와 연동하여 경보 메시지를 실시간으로 전송한다.

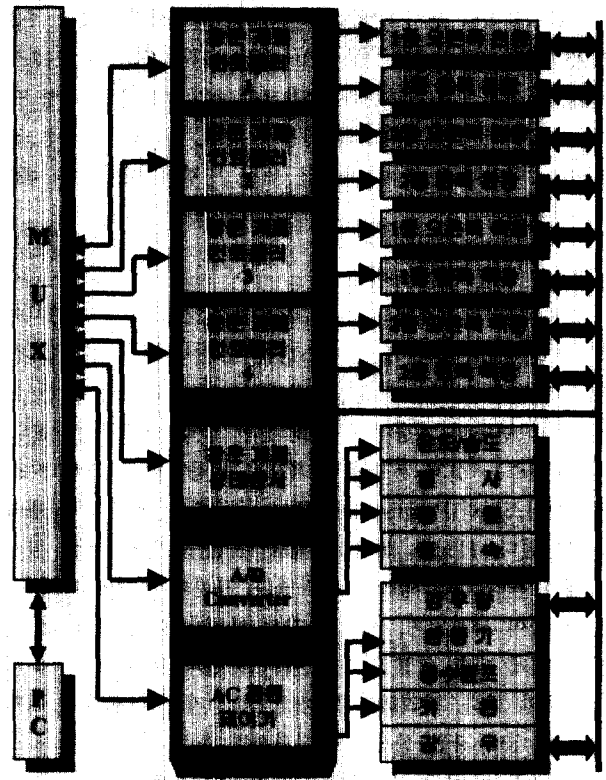
2.2 클라이언트 시스템

본 논문에서 개발된 클라이언트 시스템은 Window95/98 시분할 이벤트 복합 방식으로 단독 및 연동 온실에 대한 자원을 모니터링 및 제어한다.

2.2.1 하드웨어 컨트롤러

기존 on/off식 제어반의 컨트롤러는 전류의 흐름이 불안정하며 순간 전압의 크기와 손실이 크다. 그리고 각 시설 온실의 제어 회로는 온실 시공에 따라 회로의 크기와 형태를 수작업으로 하나의 기판에 통합 제작됨으로 회로의 확장이 불가능하고 회로의 일부 부품 파손시, 회로 사용자는 전체 제어반을 교체해야 하는 문제점을 가지고 있다. 본 논문에서 전체 온실의 각 자원을 제어하고, 모니터링하기 위해 (그림 2)와 같은 하드웨어 컨트롤러를 개발하였다.

(그림 2)의 Mux는 RS-232C 방식으로 입출력된 데이터를 PC로 전송하거나 각종 컨트롤러에서 입력된 데이터를 PC로 전송해 주는 역할을 하며 오류 데이터를 방지하기 위한 프로토콜이 내장되어 있다. 컨트롤러는 PC와의 통신을 위해 RS-232C 시리얼 포트를 사용한다. 기능적인 특성상 각 포트는 회로 상태값 저장을 위한 릴레이 포트와 온실 자원 제어를 위한 단방향 스위치 포트, 센서, on/off식 제어 패널을 쓰기 위해 제어 교환 스위치 박스를 이용하여 각 장치의 제어를 설정하는 콤보 스위치 단자를 설정해야 된다. 이러한 제어 구조는 PC나 제어반의 오작동에 대한 수동식 제어로의 전환을 용이하게 하며, 기존의 수동식 제어반에 익숙한 사용자가 시스템에 쉽게 적용할 수 있도록 한다. 또한, I/O Interface Module에서 전달되는 각 명령어에 대한 조건적 응답을 수행한다. 릴레이 포트와 스위치 포트에 대한 명령 수행의 반환값으로 대응 명령어가 반환되고, A/D 변환 포트에 대한 명령 수행 반환값은 디지털값으로 반환된다. 즉 A/D 변환 포트를 통한 센서의 전압 범위는 1V~



(그림 2) 하드웨어 컨트롤러의 구성도

<표 1> 제어 명령어 테이블

장치명	제어명령	동작	역동작	정지
1중원쪽천창	1WCA	O	C	S
1중오른쪽천창	1WCB	O	C	S
2중원쪽천창	2WCA	O	C	S
2중오른쪽천창	2WCB	O	C	S
1중원쪽축창	3WCA	O	C	S
1중오른쪽축창	3WCB	O	C	S
2중원쪽축창	4WCA	O	C	S
1중오른쪽축창	4WCB	O	C	S
온도계 1	1RAA	-	-	-
온도계 2	1RAB	-	-	-
습도계 1	1RAD	-	-	-
습도계 2	1RAE	-	-	-
풍향계	1RAG	-	-	-
풍속계	1RAH	-	-	-
일사계	2RAA	-	-	-
CO2	2RAB	-	-	-
커튼(열림)	5WCA	-	-	-
환풍기	5WCB	-	-	-
관수장치	5WCC	-	-	-
커튼(닫힘)	5WCD	-	-	-

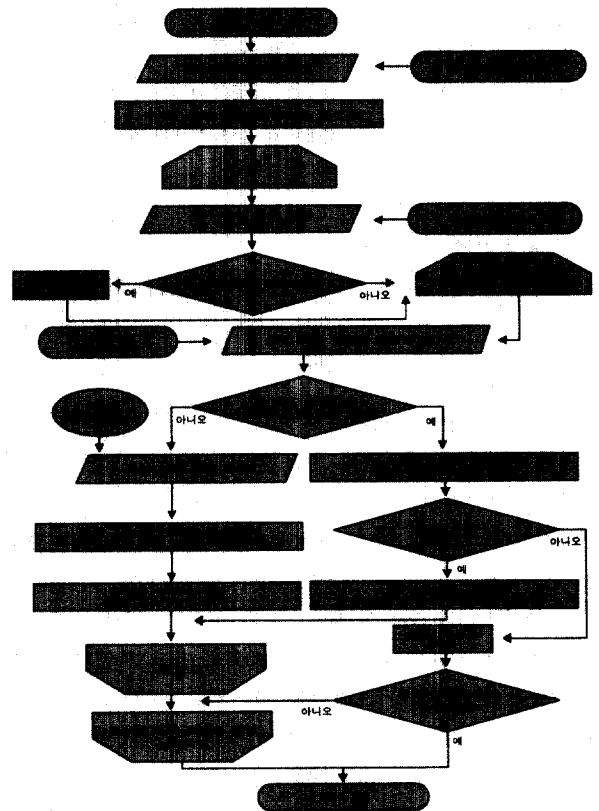
5V 사이이며, 8비트 샘플링을 통해 디지털 값으로 변환되므로 0~255 사이의 샘플링 값 범위를 가지게 된다. 따라

서, 컨트롤러의 센서 전압값을 일반적인 센서값의 환산식은 $((X1-X2)/(256-\beta)) \times (Y-\beta) + (X1+\alpha)$ 과 같고 X1: 센서 측정 최저값, X2: 센서 측정 최고값, Y: 샘플링된 측정 수치, α : 제어 인터페이스의 허용 오류값, β : 최저 측정 전압으로 나타내고, 장치 제어를 위한 명령어 테이블은 <표 1>과 같다.

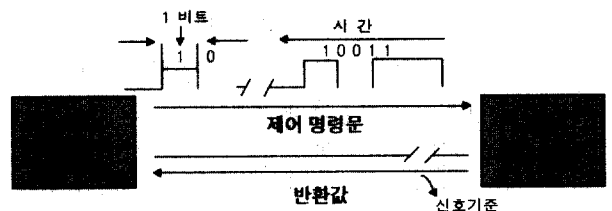
2.2.2 온실 환경 제어

하드웨어의 효율적 운용과 사용자의 신뢰성을 향상시키기 위해 온실 제어 정보 시스템은 모든 자원 측정값을 실시간에 10분 단위로 샘플링하여 분석 및 처리한다. 또한 온실의 각 단위별 계층적 분류를 통한 클래스로 구성하였다. 각 클래스는 실제 환경에 동적으로 적용시키기 위해 시스템 레지스트리(Program.ini, House.ini)를 이용하여 실행시 바운딩된다. House 레지스트리는 온실 자원에 대한 설정 상태와 관리 대상 항목을 나타낸다. 자원의 형태에 따라 설정 항목이 달라지며, 자원 추가시 House 레지스트리의 신규 자원에 대한 설정 항목을 제어 자원의 재원과 동일하게 설정해야 한다. 관리하고자 하는 신규 자원의 재원과 레지스트리 내에 자원 설정이 부정확하면 온실 환경 제어 시스템의 I/O Interface Module에서 입/출력 제어가 정확하지 않으므로 시스템이 불안정할 수도 있으니 주의해야 한다. 자동화 제어 엔진을 통해 수행되는 환경 제어 처리 흐름도는 (그림 3)와 같다. 그리고 온실 환경 제어를 위한 하드웨어를 제어하기 위해서는 직렬 통신을 이용한다. 이러한 하드웨어 컨트롤러의 통제를 위한 인터페이스 역할을 수행하는 직렬 통신은 Z-MODEM 프로토콜을 기반으로 설계하였다. 또한 직렬 통신을 위해 우선 통신 디바이스를 개방해야 한다. 통신 개방 시스템 함수는 Opencomm()이며, 통신 디바이스 개방 함수는 통신 디바이스를 열어 장치의 핸들 포인터를 반환한다. 이러한 포트 개방을 위한 포트 설정은 제어 윈도우의 'PORT' 리스트 박스를 통해 온실 자원 제어를 위한 송·수신 가능한 포트 채널을 설정할 수 있다. 통신 포트가 성공적으로 개방되면 SetCommState() 함수를 사용하여 통신 포트의 전송 속도와 데이터 비트수 등을 지정한다. 이 함수를 사용하지 않을 경우 기본적으로 OpenComm() 함수를 사용하여 2400bps, 8비트로 설정된다. I/O Interface Module의 제어 윈도우는 수신된 제어 명령에 대한 응답을 사용자 버퍼로 읽어 송신 제어 명령어의 형태에 따라 처리를 수행한다. 수신된 반환값을 받아 저장하는 큐를 읽기 위해 윈도우는 WM_COMMNOTIFY 이벤트를 발생하며, I/O Interface Module의 제어 윈도우는 이러한 수신 이벤트가 발생하였을 때, 수신 버퍼의 전체 데이터 크기만큼 버퍼의 수신 데이터를 이동시킨다. 데이터를 송신하기 위해서는 문자형 타입의 1바이트 단위의 포인터 문자 스트링을 구성하여 전송한다. 본 시스템은 직렬 통신 포트를 사용해서 데이

터를 송신하고 전송 데이터의 크기에 따른 문자 단위의 전송을 위한 시스템 함수인 TransmitCommChar() 함수를 사용하거나, 포인터 타입의 문자열을 전송하기 위해서는 WriteComm() 함수를 사용한다. 직렬 포트를 폐쇄하기 위해서는 개방시켰던 통신 포트 번호를 인자로 하여 CloseComm() 함수를 사용하여 포트를 폐쇄한다. 이러한 직렬 통신을 이용하여 제어 명령문을 처리하는 I/O 인터페이스 모듈의 직렬 통신 처리 구성도는 (그림 4)와 같다.



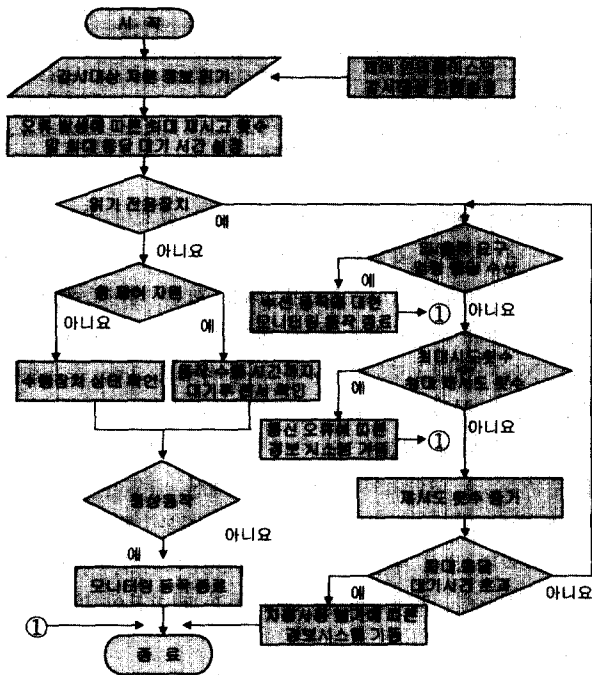
(그림 3) 환경 제어 처리 흐름도



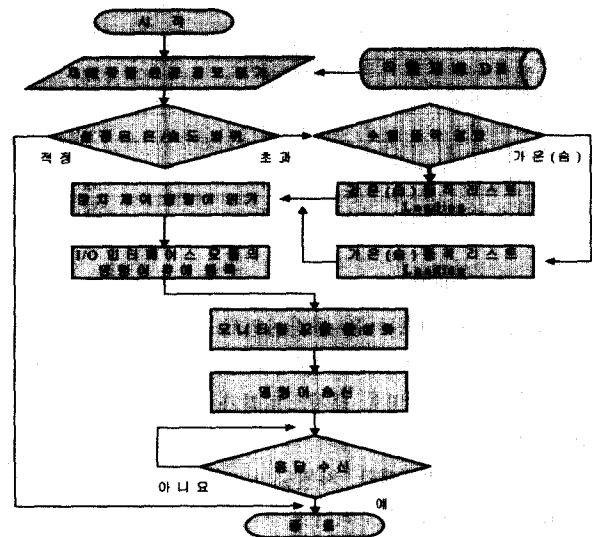
(그림 4) 직렬 통신을 위한 송수신 제어

온실 자원의 현재 상태를 감시하고 문제 발생시 적절한 대응을 수행하기 위해 온실 환경 제어 시스템은 자원 모니터링을 병행적으로 수행한다. 내·외기 환경 조건에 따른 상태 모니터링을 수행하기 위해 I/O Interface 모듈과 Monitor 모듈이 연동된다. I/O Interface 모듈은 제어반과의 직렬화된 통신을 수행하기 위해 우선 순위를 가지는 큐를

이용한다. 선형화된 동작 리스트의 각 동작이 수행될 때, I/O Interface 모듈은 Monitor 모듈을 현재 수행되는 단일 명령 동작에 할당하여 동작 감시를 수행한다. 해당 동작이 제한 시간 내에 종료되는 지를 확인하여 동작 결과에 따라 정보 시스템과 연동하여 감시를 종료한다. (그림 5)는 모니터링 모듈의 제어 처리 흐름이다.



(그림 5) 모니터링 모듈의 제어 처리 흐름도



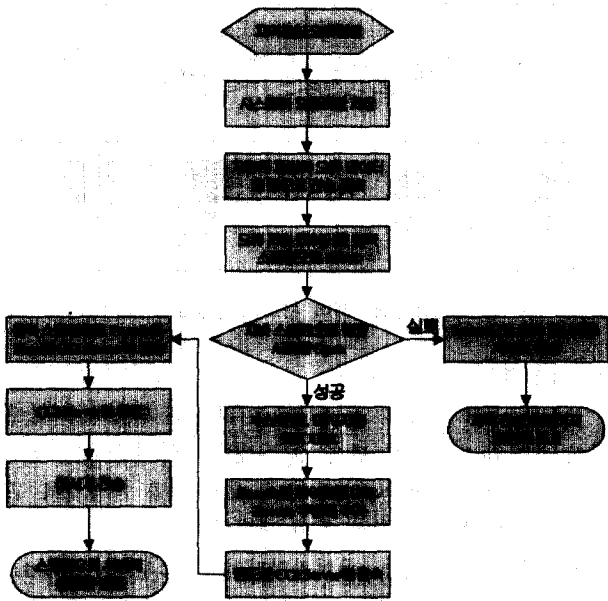
(그림 6) 온·습도 처리 흐름도

온실 제어 시스템은 운영 모드와 독립적으로 시스템에 할당되어진 현재 각 온실의 온·습도 상황을 주기적으로 점검한다. 이러한 온실 자원의 상태값을 읽기 위해서는 온

실의 하드웨어 단말 자원인 각 센서와 온실 환경 제어 정보 시스템의 소프트웨어 모듈인 I/O Interface Module간의 통신을 통해 값을 얻을 수 있다. (그림 6)은 I/O Interface Module에 의해서 선택된 온·습도 제어 처리 동작을 흐름도로 나타낸 것이다. 내기 온·습도 상황과 함께 온실의 신뢰성과 효율적 환경 제어를 위해서 외기 환경 측정은 매우 중요하다.

2.2.3 정보 메시지 전송

각 대상 온실의 관리자 부재시 대상 온실을 감시하는 클라이언트는 시스템 오류나 하드웨어적인 오류에 따른 오류 이벤트를 발생한다. 인터넷 기반의 온실 환경 제어 정보 시스템은 관리자에게 메시지를 통해 오류 발생 상황을 전달한다[5,6]. 정보 메시지 전달을 위해 시스템은 각 통신사의 SMS 게이트웨이와 연동하여 처리한다. Java의 Native method인 HttpURLConnection을 사용하여 SMS 게이트웨이에 접근하여 해당 CGI 서버로 접속한다. 정보 메시지는 오류 발생시 컴퓨터의 화면, 스피커, 핸드폰으로 오류 발생 장치를 음성 및 Text로 동시에 전송되며, 핸드폰으로 오류 메시지의 전송 방법은 (그림 7)과 같다.



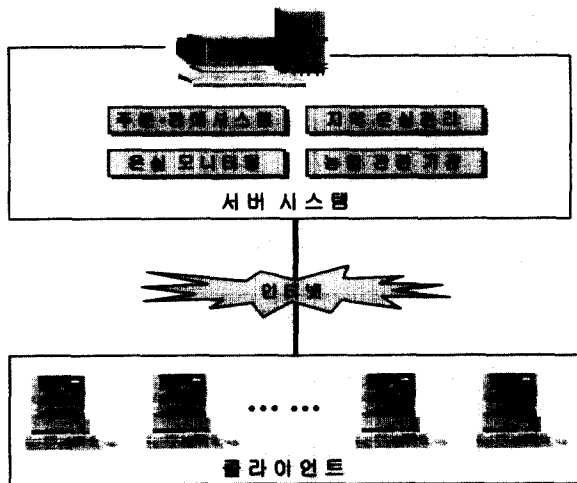
(그림 7) 정보 메시지 흐름도

2.3 서버 시스템

기존 시스템의 단순한 GUI를 통한 정보 제공 방식과 하드웨어적인 결함으로 인하여 농작물의 피해를 최소화하기 위한 단일 클라이언트 모니터링의 시간적, 공간적인 한계를 극복하기 위해 인터넷에 기반한 실시간 서버 모니터링과 정보의 시각적 표현에 중점을 두었다. 또한 지역별 온실과의 통신이 가능하며 주문·판매 시스템을 웹 서버에 통합·제공하여 재배 작물의 매매가 가능하도록 구성함으로써

써, 사용자가 인터넷을 통해 관련 작업을 효율적으로 이용할 수 있는 정보 시스템을 구성하였다.

웹 서버 구축은 NT Server 운영 체제를 사용하고 NT 운영 체제에서 최적의 성능을 나타내는 ASP를 사용하여 웹 서버를 구축하였다. 데이터 저장소로는 MS-SQL7.0을 사용한다. 또한, 플랫폼에 독립적인 자바를 사용하여 웹 서버에 온실 관리 및 온실 모니터링을 추가하였다. 클라이언트 시스템과는 별도로 서버 시스템에서는 상품을 주문하고 직접 판매 할 수 있는 주문·판매 시스템과 각 지역별 온실의 등록 상품을 관리할 수 있는 지역 온실 관리 메뉴와 원격지의 환경 정보 데이터들을 모니터링 할 수 있는 모니터링 메뉴, 마지막으로 농업에 관련된 기관을 하이퍼링크 형식으로 제공하여 운용이 미숙한 사용자들에게 편의를 제공한다. 이러한 서버 시스템 구성도는 (그림 8)과 같다.



(그림 8) 서버 시스템 구성도

2.3.1 주문·판매 시스템

농산물 유통 구조의 문제점과 어려움을 해결하기 위해 분산 네트워크 상에서 단순 정적 데이터를 제공하는 것이 아니라 농산물의 생산 과정, 크기, 모양 등을 제공하여 구매자의 제품 문의를 영상 등의 멀티미디어 정보를 이용하여 설명할 수 있는 하이퍼미디어 시스템(Hypermedia System)을 사용한다[12]. 그러므로 정보를 추출하여 의미적 순서로 구조화시키는 작업이 필요하고, 이러한 작업을 효율적으로 하기 위하여 윈도우 NT 환경하에서 최적의 성능을 가지는 ASP, MS-SQL7.0을 사용하여 DB를 구성하였다.

• DB 구성

주문·판매 시스템의 개발에 필요한 DB구성으로는 회원 테이블, 판매자 물품 등록 테이블, 임시 저장 테이블, 주문 테이블, 지역별 온실 주문 물품 임시 저장 테이블, 지역별 온실 주문 테이블, 지역별 온실 물품 등록 테이블 7개 테이블

블로 나타낼 수 있다. <표 2>는 회원 테이블을 나타낸다.

<표 2> 회원 테이블

Table Field	데이터 형	크기
일련번호	int	4
성 명	vachar	20
나 이	int	4
성 별	vachar	40
아 이 디	vachar	40
비밀번호	vachar	20
e-mail	vachar	100
주 소	vachar	255
전화번호	vachar	100
등록날짜	vachar	255

<표 2>는 회원 정보 테이블이다. 일반 고객이 처음 접속하게 되면 기존의 회원인지 일반 고객인지를 구분하고 인증이 되면 고객의 ID와 PassWord를 회원 테이블에서 확인한다. Table Field 중의 일련 번호는 속성을 IDENTITY (1,1), NOT NULL로 설정하여 자동적으로 1씩 증가하며 NULL 값을 설정하지 않는다. 주문 테이블은 <표 3>과 같으며 장바구니에 제품을 넣은 후 주문 이벤트가 발생했을 때 주문자 및 배달처와 함께 주문 계산 내용이 저장되는 테이블이다. 또한 판매자 물품 등록 테이블, 임시 저장 테이블, 지역별 온실 주문 물품 임시 저장 테이블, 지역별 온실 주문 테이블, 지역별 온실 물품 등록 테이블이 있다.

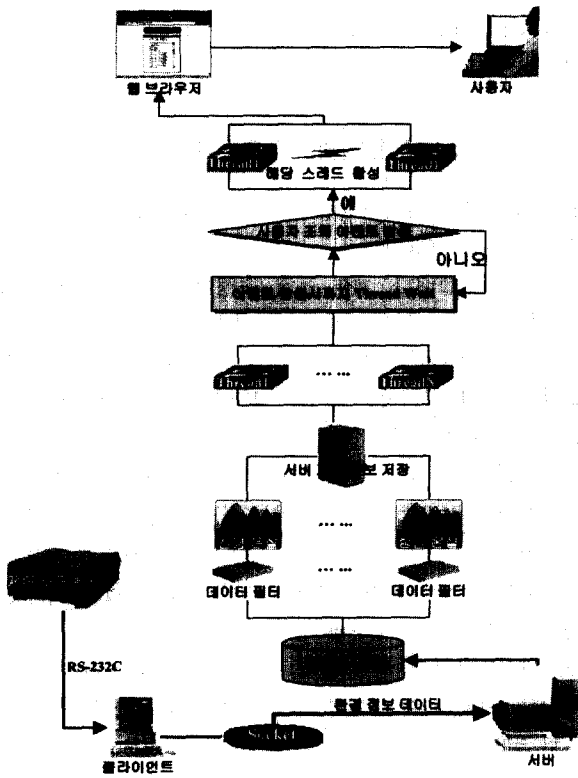
<표 3> 주문 테이블

Table Field	데이터 형	크기
일련번호	int	4
주문자 ID	vachar	20
주문자 명	vachar	40
배달전화번호	vachar	50
핸 드 폰	vachar	50
구입물품요약	vachar	255
e-mail	vachar	100
주문합계금액	int	4
상품코드	vachar	100
배달주소	vachar	255

2.3.2 서버 모니터링

각 온실 환경 자원들은 클라이언트 시스템에 의해 통제되며, 클라이언트 시스템에 제공되는 온실 자원 모니터링 정보를 기반으로 서버의 레지스트리가 구성된다. 즉 소켓을 통해 수신한 클라이언트와의 약속된 패킷이 도착하면 온실 위치 필드를 통해 해당 자원들의 값들을 자바의 Native Method인 indexof 프로퍼티를 사용하여 Filter하게 된다. 또한 패킷들을 순서적으로 정렬하기 위하여 Buffer Stream을 사용한다. 그리고 Data의 마지막까지 반복문을 수행하면서 Filter된 내용을 List Box에 표시한다. 즉 지역과 회원명, 그리고 회원에 소속된 동으로 데이터를 필터한다. 그리고 필터된 데이터는 각각의 레지스트리에 저장한 후, 동마

다 스레드를 할당한다. 온실별 환경 정보는 자바 Applet에서 이벤트를 발생하여 해당 Application을 활성화시키고 Application을 통해 클라이언트 사용자에게 보여준다. 또한 각 지역에 대한 사용자 조회 이벤트가 발생할 때 해당 스레드가 Runnable 상태로 전환하여 해당 환경 정보 데이터를 화면에 표시한다. 즉 스레드가 활성화됨으로써 java native method인 URL class의 OpenStream() 메소드를 사용하여 서버 레지스트리 파일의 선택 온실의 환경 정보 데이터를 접근한다. 또한 인터넷을 통해서 원격지 온실의 환경 정보를 확인하기 위해서 java Applet을 사용했으며 클라이언트/서버 형태로 Java native method를 이용하여 클라이언트/서버의 중앙 집중식 모니터링 방식을 적용하였고 지역 제한적 모니터링 시스템의 한계를 극복할 수 있다. 웹 서버에서의 온실별 환경 정보의 처리 구성도는 (그림 9)와 같고, 클라이언트에서 소켓을 통해 전송되어 오는 환경 정보 데이터는 (그림 10)과 같다.



(그림 9) 온실별 환경 정보 처리 구성도

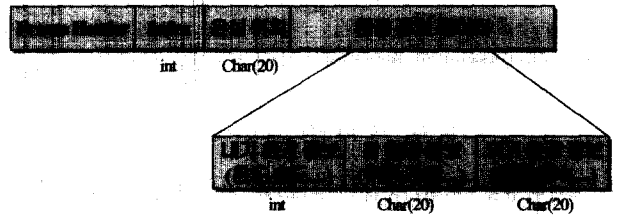


(그림 10) 환경 정보 데이터

서버 모니터링은 각 지역별 온실 정보를 소켓으로 받아 처리한다. 각 지역별 온실 클라이언트가 보내오는 환경 정보 데이터를 소켓으로 받아 시스템 레지스트리에 저장하고, 서버의 모니터링 데이터를 처리하기 위한 모니터링 표시

인터넷 애플리케이션이 모니터링 정보를 참조하여 화면에 표시한다. 모니터링을 위한 데이터 프레임 중 초기 연결 프레임은 소켓 연결 후에 처리되며, 현재 온실의 인덱스, 온실의 위치 단위로 프레임을 필터하여 모니터링 서버의 레지스트리에 저장된다. 환경 정보 데이터 필드에서 내기 환경 정보는 각 지역 온실의 내부 장치 및 센서 측정값으로 구성되며, 센서 측정값은 int형 정수값으로 나타내어 지고, 자원 제어 장치는 bool형으로 구성되어 장치 구동 및 정지 상태를 나타낸다.

각 창 의 상태를 나타내는 창 환경 정보는 온실의 각 창 의 상태를 나타내는 프레임으로 온실을 구분하기 위한 온실 인덱스만이 int형 정수로 나타내어지고, 이외의 구성 요소는 광센서의 측정값을 나타내는 bool형 이진값으로 표시된다. 외기 환경 정보는 온실 외부의 기상 상태를 측정하기 위한 센서들의 측정값으로 프레임을 구성하며, 감우 센서만이 bool형으로 우천 여부를 나타낸다. 이외의 모든 센서 측정값은 int형의 정수값으로 취급된다. 이러한 처리를 위한 전송 데이터 프레임은 (그림 11)과 같다.



(그림 11) 전송 데이터 프레임

3. 구현 및 결과 분석

3.1 구현 환경

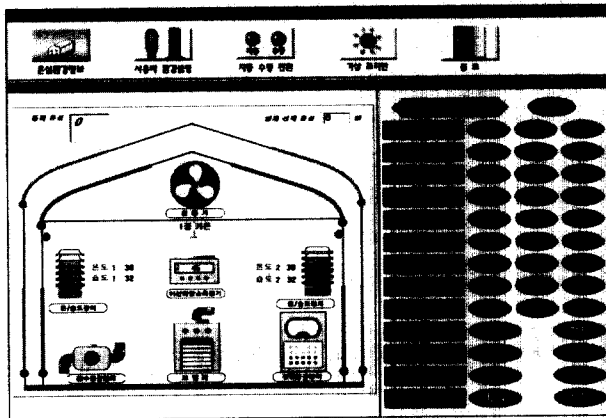
본 논문에서 구현한 인터넷 기반의 온실 환경 정보 시스템은 클라이언트와 서버 환경에서 운용된다. 클라이언트 시스템의 구현 환경은 DB 구축을 위한 InterBase 5.0, 시스템 구축을 위한 BC++3.0, 그리고 윈도우 98 환경하에서 운용될 수 있도록 구성하였다. 또한 시분할 방식으로 대상 온실에 대한 자원을 제어 및 처리하고, 재배 작물에 대한 정보를 사용자가 손쉽게 조회할 수 있는 통합 관리 시스템으로 구현된다. 서버 시스템의 운영 체제로는 NT Server4.0을 사용하며 운영 체제와 호환이 잘 되는 ASP를 사용해 웹 페이지를 구성하였고, IIS(Internet Information Server)4.0을 이용하여 인터넷 서비스를 한다. 또한, 각각의 상품 정보와 사용자 관리를 위하여 데이터베이스로는 SQL-Server7.0을 사용한다. 그리고 인터넷을 이용하여 원격지에서의 대상 농가의 온실 상황을 모니터링할 수 있도록 하였고, 운영 체제에 독립적인 Java를 사용하여 단·연동 온실의 환경 정보와 온실별 통신, 접속된 온실 정보를 인터넷상에서 관리 및 모니터링할 수 있다.

32 시스템 구현

개발된 시스템은 클라이언트/서버 기반으로 구현되었다. 클라이언트의 모니터링 윈도우와 온실의 오류에 대한 경보 메시지로 구성된다. 서버 시스템은 윈도우 NT로 구축한 웹 서버에 독립적으로 구현되어 있는 주문·판매 시스템, 서버 모니터링, 지역별 온실과의 통신으로 구성된다.

● 클라이언트 모니터링 윈도우

온실 내·외부 자원 모니터링 상태는 (그림 12)와 같이 나타낸다.



(그림 12) 주 메뉴와 모니터링 윈도우

모니터링을 위한 동작은 시스템 가동시 초기의 상태는 시스템 레지스트리(Program.ini, House.ini)를 이용하여 초기의 온실 모니터링을 화면에 표시한다. 사용자의 편의성을 제공하기 위하여 왼쪽 창의 각 장치를 터치하는 것과 오른쪽 창의 각 버튼을 이용하여 각 장치의 동작을 병행 수행할 수 있도록 구현하였다. 온실의 모든 창을 제어하기 위해 창 제어 모터가 설치되어 있고 표시 가능한 모든 자원을 모니터링 가능하며, 이외의 모니터링 대상 자원들은 각 장치 측정값을 장치 이미지와 함께 표시하고 있다. 또한 모니터링 윈도우는 현재 온·습도 등의 온실 내부 환경 정보 및 현재 자원 상태를 사용자가 이해하기 쉽도록 이미지들로 구성되어 있으며, 현재 모니터링 상태를 실시간 이미지 교체를 통해 쉽게 인지할 수 있다. 각 이미지 교체를 위한 시간 간격은 장치 클래스 내의 전체 동작 시간에 이미지 개수를 나눈 시간으로 설정된다. 이러한 교체 시간 간격은 동작 버튼 윈도우의 각 장치 수행 및 정지시 동적으로 계산되므로 각 장치 애니메이션을 위한 이미지를 독립적으로 관리할 수 있는 장점이 있다.

상단의 메뉴 윈도우는 온실의 각 장치 제어를 위한 현재 설정 상태와 시스템 운영 모드, 외기 환경 정보 조회 등을 제공하기 위한 기능 윈도우들의 링크를 제공한다. 즉 온실 환경 정보 메뉴는 각 작물의 종류에 따른 재배 환경 정보를 모니터링하며 각 장치들을 제어할 수 있다. 사용자 환

경 설정 메뉴는 각 온실의 환경을 사용자가 임의로 설정할 수 있다. 자동·수동 전환 메뉴는 시스템을 자동, 수동으로 설정하는 메뉴이며, 기상 표지판 메뉴는 현재의 기상 상태를 각 센서들의 값으로 알 수 있다. 종료 메뉴는 시스템을 종료시킨다. (그림 12)의 온실 모니터링 창에서 1중 왼쪽 측창에 대한 제어와 이미지 교체 시간 간격을 계산하기 위한 소스 코드를 (그림 13)에 나타내고 있다.

```
Tw_ConditionInformation::ci_oSWL1Click
{
    ... //제어 동작 수행 w_SystemInformation->ct_LeftSideWin1->
    Hint = "1중 왼쪽 측창 개방";
    //동작 수행 시간 간격 설정 w_SystemInformation->ct_LSWin1->
    Interval = (w_SystemInformation->Window
    [w_Logo->system.HouseSelected].DExT[5][0] * 1000)
    Window[w_Logo->system.HouseSelected].
    Window[WL1].ImageNum;
    //개방 수행 플러그 w_SystemInformation->Window[w_Logo->
    system.HouseSelected].
    Command[5] = 1; //타이머활성화
    w_SystemInformation->ct_LSWin1->
    Enabled = true; //제어 코드수행
    w_Interface->Exec(w_Logo->
    system.HouseSelected, 4, 1);
    //1중 왼쪽 측창 열기...
}
```

(그림 13) 1중 왼쪽 측창 개방을 위한 소스 코드

1중 왼쪽 측창 개방을 위해 Tw_ConditionInformation 클래스의ci_oSWL1Click

메소드가 호출된다. 제어 동작 수행에 대한 이미지 텍스트를 출력한 후, 동작 수행 시간 간격을 설정한다. 모니터링 윈도우의 DExT 멤버 변수는 전체 수행 동작 시간을 나타내며, 현재 이미지의 개수를 Window 클래스의 ImageNum 멤버 변수로 나타낸다. 이 두 멤버 변수를 나눈 값을 이미지 교체 타이머의 시간 간격으로 설정하기 위해 1중 왼쪽 측창 윈도우의 동작을 위한 타이머인 ct_LSWin1의 Interval 멤버 변수에 저장한 후, 타이머를 활성화시켜 제어를 수행한다. 장치 동작과 모니터링을 위한 타이머를 활성화한 다음에 제어 윈도우에 창 개방을 위한 제어 명령어를 Exec 메소드를 이용하여 명령 수행 큐에 등록하여 처리를 수행한다. 이외의 각 장치의 시간 간격과 동작은 위와 같이 유사한 방법으로 처리된다.

● 제어 시스템의 자동화 제어 엔진

자동화 제어 엔진은 온실 환경 제어 시스템의 환경 자동 제어와 작업 스케줄링과 모니터링을 병행하여 통제한다. 자동 모드 설정시 온실 환경 제어 엔진은 시스템의 모든 사용 권한을 가지며, 필요시 실행 시간에 운영 환경 조건을 변화시킬 수 있다. 자동화 제어 엔진의 환경 제어 동작은 명령어의 리스트로 미리 정의되어 있으며, 각 자원의 우선 순위에 따라 처리된다. 자동화 제어 엔진은 환경 제어 동작 수행을 위해 기능에 따른 미리 정의된 우선 순위를 가지는

동작 리스트를 가지고 있다. <표 4>는 이러한 동작 리스트와 우선 순위를 나타낸다.

<표 4> 엔진 제어 동작 리스트와 우선 순위

동작명	동작 순서	동작 정의
가온순서	1. 커튼 폐쇄 2. Fan 정지 3. 축창 폐쇄 4. 천창 폐쇄 5. 보일러 작동	온도 변화에 비교적 적은 영향을 받는 자원 순서로 제어
감온순서	1. 보일러 작동정지 2. 천창 개방 3. 축창 개방 4. Fan 작동 5. 커튼 개방	급작스런 온도 변화를 우려한 감시 제어 필요
가습순서	1. 가습기(Mist) 작동 2. 커튼 폐쇄 3. Fan 정지 4. 축창 폐쇄 5. 천창 폐쇄	습도 변화에 비교적 적은 영향을 받는 자원 순서로 제어
감습순서	1. 가습기 작동 정지 2. 천창 개방 3. 축창 개방 4. Fan 작동 5. 커튼 개방	급작스런 습도 변화를 우려한 감시 제어 필요

```

class Engine
{ private: //제어 동작 리스트
  struct JobStore_ Job[MAXACTION];
  bool __fastcall LoadJob();
  struct resState_ resState; //Engine Resource
  void __fastcall setResRange();
  AnsiStringDetermineAction ();
  void __fastcall beginJob();
  void __fastcall saveAction();
  void __fastcall setOutResRange(); //외기 환경
  AnsiString AnalyWindow();
  int __fastcall readState(); //동작을 구조체에 저장
  void __fastcall Force();
  __fastcall Engine(); };
    
```

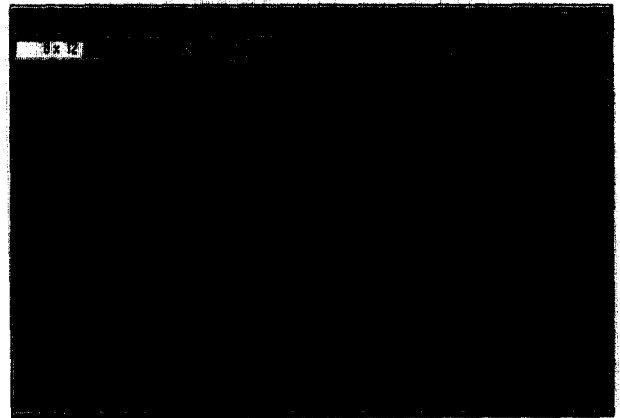
(그림 14) 환경 제어 엔진 클래스

제어 엔진 클래스는 다동 온실 제어를 위해 시스템 독립적 클래스로 설계되었다. 엔진은 각 온실에 대해 매핑되며, 할당된 대상 온실 자원의 상태값을 저장하기 위한 맴버 클래스와 메소드로 구성된다. 환경 자동 제어를 Main System Module은 각 온실을 제어하기 위해 엔진을 순차적으로 할당한다. 각 엔진의 제어 순서는 Main System Module에서 할당된 인덱스에 따라 처리되며, Main System Module의 엔진 문맥 교환(Context Switch)은 각 온실에 대한 레지스트리 파일에 의해 처리된다. (그림 14)는 환경 제어 엔진 클래스를 나타내고 있다.

● 정보 시스템 기능

휴대폰을 이용한 정보 메시지 전달을 위해 무선 문자 메시지(Short Message Service) 서비스를 이용하였다. SMS 서비스는 디지털 이동 전화의 액정 화면(LCD)에 한글 40자

내외의 간단한 문자 메시지(영문 80자)를 인터넷이나 휴대 전화로 전송할 수 있는 서비스이다. 정보 시스템은 신세기 통신의 SMS 게이트웨이와 연동하여 정보 메시지를 처리한다. (그림 15)는 핸드폰 전송 후 수신된 응답 화면이다. 즉 정상적인 송신이 되었을 때 SMS 게이트웨이의 송신 응답 화면이다. 그리고 송신 응답을 받은 후 실제로 사용자에게 전송되는 핸드폰 오류 화면은 (그림 16)과 같다. 내용은 오류 메시지 내용과 발신지가 수신된다.



(그림 15) 핸드폰 전송 후 수신된 응답 화면



(그림 16) 핸드폰에 전송된 오류 화면

(그림 17)는 정보 메시지의 처리 코드이다. 즉 신세기 통신의 CGI 서버에 접속한 후, 출력 스트림을 생성하여(OutputStreamWriter) 해당 정보를 SMS 게이트웨이에 Encode 형식으로 송신한다. 정상적인 송신이 되었을 때 입력 스트림을 생성하여(InputStreamReader) SMS 게이트웨이의 응답 메시지를 수신하게 된다.

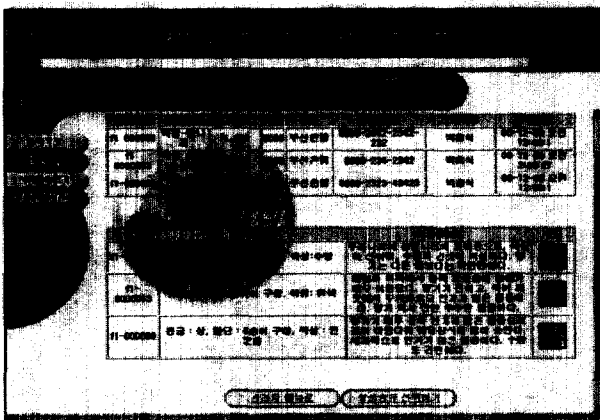
● 주문·판매 시스템 윈도우

주문·판매 시스템은 ASP를 이용하여 웹 서버로 구성된다. 이는 생산과 판매 체계 기능을 하나로 통합하여 신속하고 투명한 생산물 유통 경로를 확립할 수 있다. (그림 18)은 서버에 구축되어 있는 주문·판매 시스템의 관리자 메

뉴스에서의 등록 상품 윈도우이다.

```
protected Stringserver_url =
"http://messenger.shinsegi.com/cgi-bin/
messenger/messenger_sub11.0602.blocking";
//신세기통신의 SMS게이트웨이로 접근
httpURLConnection =
(new java.net.HttpURLConnection)
url.openConnection(); //출력스트림생성 .....
java.io.PrintWriter pw =
new java.io.PrintWriter(osw, true);
//필요한 파라미터값 전송, 핸드폰 번호 입력
pw.print("phonol = " + java.net.URLEncoder.
encode(this.to_number) + "&");
pw.print("phono2 = " + java.net.URLEncoder.
encode(this.to_vender) + "&"); ...
//SMS게이트웨이의 응답을 받기 위한 입력 스트림 생성
java.io.BufferedReader br =
new java.io.BufferedReader
(new java.io.InputStreamReader
(httpURLConnection.getInputStream()));
//응답 내용을 수신하기 위한 버퍼 생성
StringBuffer buf = new StringBuffer();
String line = "";
//응답 스트림을 한 행씩 읽어옴
while( (line = br.readLine()) != null )
buf.append(line + "\n");
```

(그림 17) 경보 메시지 처리 코드

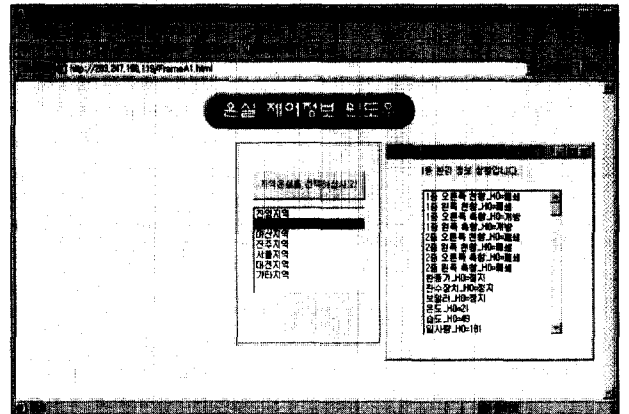


(그림 18) 등록 상품 윈도우

● 지역별 온실 모니터링 윈도우

지역별 온실 환경 정보를 인터넷을 통하여 서버에 등록이 되어 있는 온실을 실시간으로 모니터링할 수가 있다. 소켓으로 원격지의 환경 정보 데이터를 정의된 프레임(그림 11)로 수신하며 수신된 프레임을 서버 파일 레지스트리에 저장한다. 저장된 환경 정보 데이터를 각 지역, 동에 적합한 데이터를 화면에 표시하기 위하여 수신된 데이터에 대한 Filter를 수행하게 된다. 자바 Applet을 사용하여 현재 등록되어 있는 등록 온실 리스트의 윈도우가 할당된다. 각 리스트를 이벤트에 등록하여 해당 온실을 선택시 그 온실에 대한 이벤트가 발생하여 해당 온실에 대한 프레임을 생

성한다. (그림 19)는 지역 온실 리스트에서 선택한 온실(함안 지역의 1동 온실)의 정보에 대한 환경 정보 데이터를 실시간으로 전송받아 화면에 표시하고 있다.



(그림 19) 환경 정보 상황

원격지 클라이언트의 환경 정보 데이터를 수신하기 위하여 ServerSocket를 생성하고 약속된 프레임이 서버로 전송되면 프레임 필드중에 온실 위치 필드를 filter를 하여 해당 서버 레지스트리에 저장한다. 이러한 과정을 나타내는 Data Filter에 대한 처리 코드는 (그림 20)과 같다.

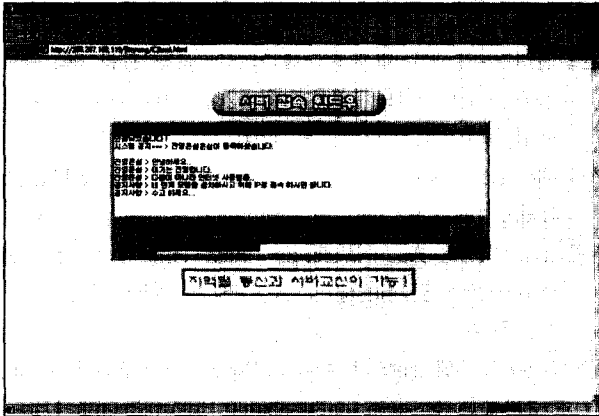
```
public class Server { .....
ServerSocket InfoSocket = null;
.....
try { //서버 소켓 생성
InfoSocket = new ServerSocket(port);
} catch(IOException e) { ..오류 처리.. }
.....
while(listening) { // 다중 스레드 실행
new ServerExeThread(
InfoSocket.accept().port).start();
.....
public void Filter() { // 필터를 처리
String filterdata;
int total = Frame.length();
//프레임의 길이
//지역 필드를 구분하는 "_" 검색
int diff = Frame.indexOf("_");
if(0 < diff) {
filterdata = Frame.substring(diff+1, diff+3);
if(filterdata.equals("J0")) { // 해당 동을 필터
JinoutputFile1.println(inLine); }
// 레지스트리 저장
.....
}
```

(그림 20) Data Filter의 처리 코드

● 지역별 온실과의 통신 윈도우

네트워크 단말 노드인 지역 온실이나 온실 관리 시스템 서버와의 연결이 필요할 때 웹 서버를 통한 연결 서비스를 제공한다. 연결된 각 온실 클라이언트는 자바 Applet의 소켓을 이용하여 서버에 등록된다. 그리고 온실 관리 시스템 서버는 각 온실 클라이언트의 연결 시간, 연결 경로를 온실

관리 시스템 GUI에 표시된다. (그림 21)은 지역 온실과의 통신 윈도우이다.



(그림 21) 지역 온실 통신 윈도우

지역 온실 통신 윈도우를 이용하여 사용자는 각 지역 온실 및 온실 관리 시스템의 서버를 통해서 통신 서비스를 제공받을 수 있다. Applet의 이벤트가 발생하면 actionPerformed() 메소드가 실행된다. 이 메소드에서는 서버의 IP와 port를 할당하고 새로운 프레임 jinCommu Application을 생성한다. 이 프레임은 통신 모듈 및 UI 등 핵심적인 처리를 하는 프레임이다. 지역별 온실의 통신 접속 윈도우에 대한 처리 코드는 (그림 22)와 같다.

```

Class EventB implements
ActionListener { // 이벤트 발생
void actionPerformed(ActionEvent e){
.....
serverIP = "203.247.166.119";
serverPort = "2000";
jinCommu.fixServer(serverIP,serverPort);
..... }

private void setEventHandler() {
ChatEventHandler handler =
new ChatEventHandler();
chatFrame.addWindowListener(handler);
//이벤트에 등록
connectionBtn.addActionListener(handler);
.....
sendMsgFld.addActionListener(handler);
If ( source == connectionBtn ) {
connectToServer(); } else if
..... }
    
```

(그림 22) 지역별 온실 통신 접속 윈도우의 처리 코드

3.3 결과 분석

본 논문에서 개발된 인터넷 기반의 온실 환경 제어 정보 시스템은 모니터링과 제어를 위해 최적화된 자동화 제어 엔진과 이를 기반으로 한 온실 환경 제어를 수행하였고 클라이언트/서버 형태로 Java native method를 이용하여 클라이언트 지역 온실을 원격으로 관리하고 모니터링하는 방

법을 제시하고 구현하였다. 특히, 기존 연구에서 구현된 시뮬레이션 환경에서의 제어를 현장의 온실에 직접 적용하여 사용자의 신뢰도를 향상시켰다. 또한 DOS 환경하에서의 모니터링 시스템과 on/off 식의 제어반만을 이용한 시스템에 비해 효율적이며, 여러 동의 온실을 제어하는 편리성을 제공하고 있다. 특히, 제어 하드웨어와의 통신 장애와 실체 장치 수행에 대한 신뢰도를 높이기 위해 모니터링 모듈은 동작 대상 주변 장치들을 실시간 모니터링한다. 그리고 모니터링을 위해 최소한의 자원만을 활용하여 추가적 제어 자원에 대한 시스템 구축 비용을 절감하였다. 또한 기존의 시스템에서는 원격지에서 오류 발생 상태를 알 수 없고, 한 제어반에서는 여러 동의 온실을 제어할 수 없다. 그리고, 온실 환경 변화에 대한 데이터를 보관할 수 없으며, 주기식 타이머를 이용한 환경 제어 모니터링을 수행함으로써, 실시간 상황 감시가 이루어지지 않았다. 그러나 본 시스템은 기존 시스템의 문제점을 보완하여 같은 노동 인원으로 노동 시간을 증가시키지 않고 재배 면적을 40%이상(최대 255동 까지 확장 가능) 증가시켰다. 온·습도 등 매일의 데이터를 정확하게 저장함으로써 재배 기술을 보편화시켰다. 그리고 기기 자동화로 인해 관리자가 온실 지역을 벗어나도 핸드폰으로 온실의 오류 발생을 전달해 주기 때문에 사용자 편의성을 높였다. 인터넷을 이용한 지역 온실의 환경 정보 모니터링을 수행하여 원격지에서도 해당 온실의 상황을 모니터링할 수 있도록 하였다. 추가적으로 농업에 관련된 유용한 인터넷 사이트의 정보와 주문·판매를 웹 서버에 통합하여 제공함으로써 사용자들의 편의성을 제공할 수 있다. 또한 서버를 경유해 다른 지역과도 정보 교류도 가능하다. 클라이언트/서버의 중앙 집중식 모니터링 방식을 적용하여 지역 제한적 제어 시스템의 한계를 극복할 수 있으며 원격지 환경 정보를 인터넷을 통하여 모니터링이 가능함으로 항상 상주해야 한다는 불편함을 극복할 수 있다. 따라서, 개발된 인터넷 기반의 온실 환경 제어 정보 시스템을 이용하면 현재 온실의 상황을 네트워크를 통해 효과적으로 모니터링 가능하므로 사용자의 온실 관리 및 정보 획득에 대한 편의성을 제공할 수 있다. 그리고 텍스트 기반의 비효율

<표 5> 기존 시스템과의 비교

구분	기존 시스템 [3][4][5]	본 논문의 시스템
동작 수행 방식	On/Off식 제어	액정 패널 및 컴퓨터에 의한 제어
단동·연동 온실 제어	가능	가능
온실 환경 데이터의 장기 보관 및 분석	불가능	가능
핸드폰으로 오류 메시지 전송	불가능	가능
실시간 상황 감시	불가능	가능
인터넷상의 데이터 획득 및 모니터링	불가능	가능
지역 및 온실별 상호 통신	불가능	가능
주문·판매	불가능	가능
가 격	저 가	고 가

적 순차 제공 메뉴 방식을 계층적 GUI로 보완하여 컴퓨터 운용 능력이 미숙한 사용자에게 보다 쉬운 접근을 제공한다. 그리고 온실 클라이언트는 온실 관리 시스템의 웹 서버에 접속하여 온실 클라이언트들과 서버와의 정보 교류 및 통신이 가능하다. 기존 연구의 시스템과 기능 비교를 <표 5>에서 요약하고 있다.

4. 결 론

국내의 시설 원예 농가에도 인터넷의 보급이 확산되면서 인터넷에 접목을 하려는 시도가 증가하고 있다. 그러나 대부분의 농가들은 영세할 뿐만 아니라 컴퓨터의 운용 능력이 미숙하며 컴퓨터를 이용한 정보 이용의 혜택을 누리지 못하고 있다.

따라서, 본 논문에서는 인터넷을 이용하여 시간과 장소에 구애받지 않고 정보를 공유할 수 있도록 하고, 사용자들의 편의를 제공하기 위하여 계층적으로 구현하였다. 또한, 기존 시스템에 국한되어 있는 시간적, 공간적 기능의 한계를 탈피하고, 웹 서버로서의 기능과 인터넷의 목적을 최대화하였다. 시뮬레이션 수행시 발생하는 동작의 추상성과 제한된 시뮬레이션 환경 조건을 실제 현장에서 적용한 결과를 비교 분석하여 보완하였다. 특히, 제어 수행시 발생하는 실제 장치 오류와 각 상태에 대한 모니터링에 의한 다음 동작 판단을 위해 내·외기 환경 정보를 실시간으로 구성하여 판단하므로, 시스템의 안전성을 크게 높였다. 또한 인터넷을 이용하여 각 환경 장치들을 모니터링이 가능하므로 원격지에서라도 자신의 온실 모니터링이 가능하다.

향후의 연구 과제에서는 서버 모니터링이 일반 텍스트가 아닌 실제 VOD로 모니터링하는 시스템과 모든 핸드폰의 경보 메시지 전송 기능을 구체적으로 설계하고자 하며 인터넷을 이용한 실시간 원격 제어를 위한 시스템을 개발하고 지역의 균형 발전을 위해 농가에서도 인터넷을 이용한 정보를 획득함으로써 첨단 과학적 영농을 도모한다.

참 고 문 헌

[1] 권영삼, "국내 원예시설의 유형별 특성과 발전 방향", 국내의 원예 시설의 특성과 시스템에 관한 심포지엄, 한국시설원예연구회, pp.111-125, 1995.
 [2] 민영봉, 박중훈, 이상욱, 정대상, "온실의 복합환경제어 시스템의 구성", 경상대학교 시설 원예 연구, pp.151-165, 1994.
 [3] 김문기, "마이크로 컴퓨터를 이용한 온실 환경 제어", 경상대학교 시설 원예 연구, pp.261-331, 1994.
 [4] 홍순호, "Automatic Control of Growth Environment for

Plant Factory", 서울대학교 박사 학위 논문, 1995.

[5] 서상진, 박홍식, 박홍복, "온실 환경 자동화 시스템의 설계 및 구현", 한국정보처리학회, 제12회 추계학술대회, p.463, 1999.
 [6] 김대업, 서상진, 박홍복, "웹 기반의 온실 관리 시스템의 설계 및 구현", 인터넷정보학회 제2회 추계학술발표대회는 문집, 제1권 2호, pp.32-36, 2000.
 [7] Merlin Hughes, Michael Shoffner, Derek Hamner, Umesh Bellur, "java Network Programming," Manning, 2000. 5.
 [8] Alan Burns, Andy Wellings, "Real-Time Systems and Programming Languages," ADDISON-WESLEY, pp.95-125. 1996.
 [9] Craig M. Wittenbrick, Eric C. Rosen, Darrell D. E. Long, "Real-time System for Managing Environmental Data," Proceeding of Conference on Swotware Engineering and Knowledge Engineering, June 1996.
 [10] Bamshad Mobasher, Robert Cooley, Jaideep Srivastava, "Automatic personalization based on Web usagemining," CACM 43(8), pp.142-151, 2000.
 [11] "Providing Customers Information Using the WEB and CORBA," URL : <http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/DDay/hastings/hastings.html>.
 [12] Juliana Nelson, "Global Web Site Analysis Market Review And Forecast 1997~2002," IDC Market Research, Mar. 1998.



김 대 업

e-mail : cool7312@unicom.pknu.ac.kr
 2000년 동서대학교 컴퓨터공학(공학사)
 2000년~현재 부경대학교 대학원
 전자계산학과 석사과정



박 홍 복

e-mail : multiphb@pknu.ac.kr
 1982년 경북대학교 컴퓨터공학(공학사)
 1984년 경북대학교 대학원 컴퓨터공학
 (공학석사)
 1995년 인하대학교 대학원 전자계산학
 전공 (이학박사)
 1984년~1995년 동명대학 전자계산학과 부교수
 1996년~현재 부경대학교 전자계산학과 부교수
 관심분야 : 실시간 시스템, 멀티미디어 응용, 산업자동화, 원격
 교육, 프로그래밍 언어 및 컴파일러,