
자동 환경 제어 모니터링 시스템 개발

김대업* · 엄현서* · 정광하* · 박홍복**

A Development of Monitoring System for Automatic Environmental Control

Dae-Up Kim^{*} · Hyun-Seo Eom^{*} · Kwang-Ha Jung^{*} · Hung-Bog Park^{**}

요 약

현재 국내의 산업 자동화 관련의 온실 환경 제어 시스템은 노동 집약적 시설에 타이머를 이용하여 각 장치를 제어하는 on/off식 제어반으로 구성되어 있다. 그러나 이러한 시스템은 야간이나 원격지에 있을 경우에는 온실의 상태 파악이 어렵다. 따라서, 본 논문에서는 기존의 on/off식 제어 방식을 액정 패널로 수동 제어를 하고, 컴퓨터로 수동 및 자동 제어 상황을 모니터링할 수 있는 시스템을 개발하였다. 또한, 재배 작물에 대한 관련 정보를 DB로 구축하여 사용자에게 정보 획득의 편의성을 제공하도록 한다.

ABSTRACT

At present, environmental control system of greenhouses related with industrial automation in the korea is composed of on/off-type control machine using a timer in facilities requiring intensive labor. This system, however, makes it difficult for us to find out any status of environmental control system in greenhouses when we are situated in a remote area or at midnight. In this paper we improved a system to monitor status for manual and automatic control by means of PC by replacing an existing on/off-type controller with a display panel. It also attempts to implement DB for information on corps and to provide a user with the efficiency of getting information for plants.

1. 서 론

* 부경대학교 대학원 전자계산학과 석사과정

** 부경대학교 전자컴퓨터정보통신공학부 부교수

접수일자 : 2001년 2월 11일

생산하는 제품의 질을 향상시키고 생산 효율을 극대화시키기 위하여 인력, 자재 및 장비를 관리

하는 것도 중요하지만 생산 과정에 필요한 정보를 수집, 감시, 저장, 제어, 분석하여 효율적인 생산 시스템을 구성하고 제조 환경을 최적화시킬 수 있는 자동화 시스템의 중요성은 날로 증가하고 있다[6],[11].

그러나 다양한 산업 분야에서도 국내 농업 관련 자동화 생산 기술은 농업 선진국인 일본, 네델란드 등과 비교할 때 극히 미비한 수준이다. 농산물 시장 개방 및 자연 재해에 따른 농산품의 생산 품목과 품질의 향상을 높이기 위해 농업 관련 종사자들의 어려움이 가중되고 있다. 이에 제한된 작물 재배 환경속에서 고가의 작물 재배와 품질의 향상을 기대하는 농업 종사자들은 대부분 비닐 온실을 수작업으로 운영하여 작물을 재배하고 있다[8],[9]. 농업 인력 부족과 기술 미비를 극복하고 농업 생산품의 수량과 질을 높이기 위한 작물 재배를 위한 시설 자동화 기술이 국내에서도 지속적으로 개발되고 있다[3],[12]. 그러나 관련 정부기관 및 연구, 생산업체에서 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],[7]. 시설 면적의 대부분을 차지하고 있는 비닐 온실은 재배 관리의 생력화 정도가 낮고 노동 집약적 생산 체계이며, 광, 온도, 습도 환경이 불리하여 품질 및 생산성이 떨어지는 실정 이어서 관계 시설에 대한 시장의 크기가 협소하며 제어 자원 모니터링에 대한 관련 연구도 아직도 많이 미흡하다. 또한, on/off 식의 제어반을 사용함으로써 야간 또는 원격지에서의 온실 상황을 알 수 없는 문제점이 있다. 이러한 문제를 해결하기 위해서 기존의 연구[11],[13]에서는 시뮬레이션 환경하의 작물 생산 과정에 필요한 정보를 수집, 감시, 저장, 제어를 수행하는 효율적인 생산 시스템을 설계하였다.

본 논문에서는 기존의 연구에서 설계 및 구현된 방법을 실제 작물 재배 환경에 적합한 하드웨어 컨트롤러와 자동 환경 제어 모니터링을 위한 소프트웨어를 개발하였다. 특히, 시뮬레이션 수행

시 고려되지 못한 실제 자원의 오류를 보완하였고, 시스템의 각 제어 자원의 동작 수행을 감시하는 모니터링 모듈의 실시간 자원 모니터링을 최소한의 자원으로 운영하도록 개발하였다. 또한, 제어 인터페이스와 컨트롤러 사이의 순서화된 직렬 통신 환경에서 시스템의 다중 명령 수행 제어를 위해 우선 순위를 가지는 시스템 큐를 추가하여 다중 접근 처리를 수행하고, 현장 적용에 의해 시스템의 성능을 평가하였다. 그리고 온실의 오류 발생시 온실 내부의 스피커를 통하여 경보 메시지 전달과 동시에 핸드폰으로 오류 메시지를 전달하도록 하였으며, 기존의 연구에서 구현된 재배 작물 DB 시스템 구성 및 처리의 기능을 향상시켰다.

II. 시스템 설계

1. 시스템 구성도

개발된 시스템 구성도는 전체 모듈을 링크하는 Main System Module과 재배 작물 및 작물 환경 정보를 가지는 RDB, 그리고 웹 서비스를 제공받기 위한 Web Browser, 시스템 운영이 자동화 모드로 동작할 때, 온·습도 등의 온실 내·외기 환경을 우선 순위에 따라 각 자원 상태를 실시간으로 탐색하여, 현재 온실내의 환경에 대한 처리 동작을 결정하는 Automation Engine, 그리고 입·출력 처리를 위한 I/O Interface Module과 처리 동작을 모니터링하기 위한 Monitor 모듈로 구성되어 있다. 이외에 PC와 온실 자원을 제어하는 Controller, Greenhouse Resources 등으로 그림 1과 같이 구성하였다.

본 논문에서 실제 적용된 모니터링 모듈은 각 창에 대해 광센서와 동작 수행 제한 시간을 이용하여 수행 완료 여부를 점검함으로써, 온실의 모니터링 제어 자원의 정확성과 비용의 효율성을 고려하였다. 기존의 논문에서 구현된 온실 복합 환경 제어 시스템은 현장 적용시 실시간 제어와 현장에서 사용된 하드웨어 자원의 특성에 기인되어 시스템 구성 요소간의 결합도가 비효율적으로 설계되어 있었다. 또한, 특정 제어 자원 중 하드웨어 동작 오류 등이 발생했을 때, 결합 허용 기

능 적용의 모호성과 오류 발생 장치의 검색이 부정확하였다.

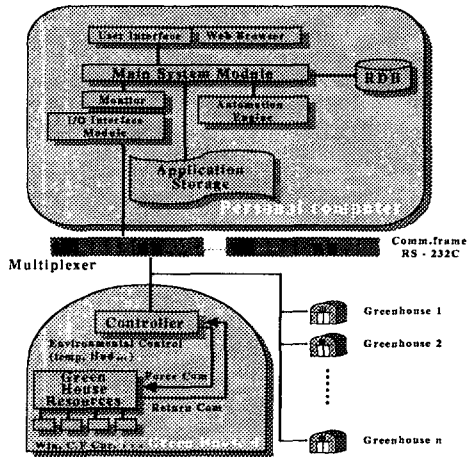


그림 1. 시스템 구성도
Fig. 1. System configuration

따라서, 본 논문에서는 시스템 처리 동작을 판단하는 Automation Engine과 하드웨어적인 입/출력 처리를 위한 I/O Interface Module, 그리고 각 동작의 모니터링을 위한 Monitor 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.

본 시스템은 온실에 대한 환경 제어를 위해 각 제어 자원에 대한 동작과 기능적 모듈들은 모두 객체화하였다. 객체 단위의 계층적 제어 구조와 객체의 기능에 따른 동작을 캡슐화하여 전체 시스템의 처리 구성과 각 객체가 사용하는 시스템 자료 구조에 일관성 있게 접근하도록 구성하였다. 실제 자원의 모니터링과 제어가 용의하고, 제어 동작의 구성을 단순·명확하게 이행함으로써, 전체 시스템 구성의 효율성과 시스템 동작 판단에 따른 신뢰도를 높였다. 그리고 현장에서 제어되는 각 동작 판단의 환경 조건을 온실 내·외기 환경을 동시에 고려하여 판단함으로써, 자원 재해에 따른 온실 피해를 최소화하였다.

그리고 내·외기 환경 조건에 따른 상태 모니

터링을 수행하기 위해 I/O Interface 모듈과 Monitor 모듈이 연동된다. I/O Interface 모듈은 제어반과의 직렬 통신을 수행하기 위해 우선 순위를 가지는 큐를 이용한다. 선형 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. 각 명령 동작은 제어반의 수행 형태에 따라 선형적이거나 병행적으로 동작한다.

2. 자원 제어를 위한 하드웨어 컨트롤러

본 논문에서 개발된 온실의 자원 제어를 위한 하드웨어 컨트롤러는 부록에 게재하였고, 그 하드웨어의 컨트롤러의 동작에 대한 블록 구성도는 그림 2와 같다.

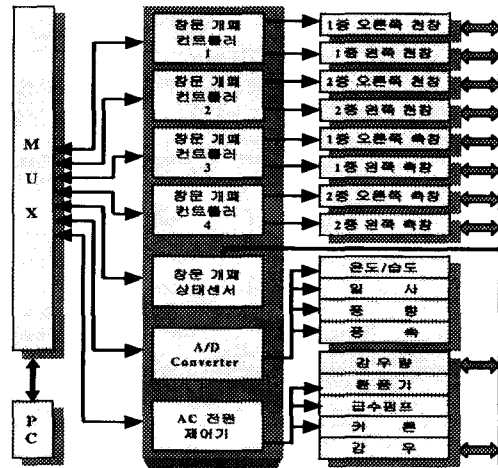


그림 2. 하드웨어 컨트롤러의 구성도
Fig. 2. Configuration of H/W controller

온실 내에서는 수동 동작을 위한 액정 패널이 있고, 야간이나 원격지에 있을 경우는 관리실의 컴퓨터에 의해 제어된다. MUX는 RS-232C 방식으로 입출력된 데이터를 PC로 전송하거나 각종 컨트롤러에서 입력된 데이터를 PC로 전송해 주는 역할을 하며 오류 데이터를 방지하기 위한 프로

도플이 내장되어 있다. MUX와 창문 모터 컨트롤러는 기능적으로 분리된 독립 회로를 통해 직접 동작이 수행되며, 각 센서 및 환풍기 등의 AC 모터 컨트롤러는 입/출력 특성에 따라 역할을 수행한다. 컨트롤러는 PC와의 통신을 위해 RS-232C 시리얼 포트를 사용한다. 기능 특성상 각 포트는 회로 상태값 저장을 위한 릴레이 포트와 온실 자원 제어를 위한 단방향 스위치 포트, 센서 입·출력을 A/D 변환 포트 구성된다.

A/D 변환 포트를 통한 센서의 전압 범위는 1V~5V 사이이며, 8비트 샘플링을 통해 디지털 값으로 변환되므로 0~255 사이의 샘플링 값 범위를 가지게 된다. 따라서, 컨트롤러의 센서 전압값을 일반적인 센서값으로 환산하기 위한 식은 그림 3과 같다.

$$(((IX1-X2)/(256-\beta)) \times (Y-\beta)) + (X1+\alpha)$$

- X1: 센서 측정 최저값
- X2: 센서 측정 최고값
- Y : 샘플링된 측정 수치
- α : 제어 인터페이스의 허용 오류값
- β : 최저 측정 전압

그림 3. 변환 포트의 센서값 환산식

Fig. 3. Sensor-value conversion expression of port

컨트롤러는 I/O Interface Module을 통해 시스템의 수행 명령을 전달받아 동작을 처리한다. 온실의 각 자원에 대한 직접적 하드웨어 처리를 위해 회로 설계시 컨트롤러와 I/O Interface Module은 상호 정의된 명령어 테이블을 통해 수행 명령을 팻치하여 RS-232C 포트를 통해 컨트롤러에게 명령을 전달한다. 표 1은 장치 제어를 위해 정의된 입·출력 제어 명령어 테이블이다. 컨트롤러는 I/O Interface Module에서 전달되는 각 명령어에 대한 조건적 응답을 수행한다. 릴레이 포트와 스위치 포트에 대한 명령 수행 반환값은 대응 명령어가 반환되고, A/D 변환 포트에 대한 명령 수행 반환값은 디지털값으로 반환된다. 명령어의 동시 입력에 대한 회로의 동작 처리는 선입선출(FIFO) 방식으로 이루어지며, 동시 명령어 입력에 대한 선입 판단이 불가능할 때에는 하나의 명령어를

폐기한다.

표 1. 제어 명령어 테이블
Table 1. The table of control command

| 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 |
| 1중원쪽측창 | 3WCA | O | C | S |
| 1중오른쪽측창 | 3WCB | O | C | S |
| 2중원쪽측창 | 4WCA | O | C | S |
| 1중오른쪽측창 | 4WCB | O | C | S |
| 온도계1 | 1RAA | - | - | - |
| 온도계2 | 1RAB | - | - | - |
| 습도계1 | 1RAD | - | - | - |
| 습도계2 | 1RAE | - | - | - |
| 풍향계 | 1RAG | - | - | - |
| 풍속계 | 1RAH | - | - | - |
| 일사계 | 2RAA | - | - | - |
| CO2 | 2RAB | - | - | - |
| 커튼(열림) | 5WCA | - | - | - |
| 환풍기 | 5WCB | - | - | - |
| 관수장치 | 5WCC | - | - | - |
| 커튼(닫힘) | 5WCD | - | - | - |

3. 작업 흐름

하드웨어의 효율적 운용과 사용자의 신뢰성을 향상시키기 위해 온실 제어 정보 시스템은 모든 자원 측정값을 실시간에 10분 단위로 샘플링하여 분석 및 처리한다.

온실의 각 단위는 계층적 분류를 통한 클래스로 구성하였다. 각 클래스는 실제 환경에 동적으로 적용시키기 위해 시스템 레지스트리(Program.ini)를 이용하여 실행시 바운딩된다. 구성 온실의 각 단위 장치의 속성은 실행 시간에 장치 리스트 구성을 위한 커스텀마이징을 유연하게 수행하기 위해 클래스의 상호 연관성을 최소화하였다. 이러한 독립적 클래스 설계를 통해 기존의 연구에서 발견된 병행 수행을 위한 동작의 의존성을 크게 줄일 수 있었고, 각 클래스의 병행 수행 능력과 시스템 운영의 효율성을 높일 수 있었다. 이러한 클래스 기반의 인스턴스로 구성된 시스템의 총괄 작업 흐름도는 그림 4와 같다.

기존 시스템의 대부분은 온실의 운영을 위한

자동·수동 운영 모드를 가지고 있으나, 기능적으로 분기식 제어에 그 기능이 제한되어 있다. 본 논문에서는 각 단위 동작을 수행하기 위한 분석 모듈과 온실 제어가 가능한 수동·자동 운영 모드를 제공한다. 특히 자동 환경 제어를 위해 독립적으로 설계된 자동화 제어 엔진을 통해 시스템 자원을 효율적으로 스케줄링하며 통제 가능하다. 온실 환경 제어를 위한 자동화 제어 엔진은 범용적 하드웨어 운용 소프트웨어 모듈로써 시스템 구축과 디버깅이 용이하여 시스템의 신뢰성을 향상시킬 수 있다.

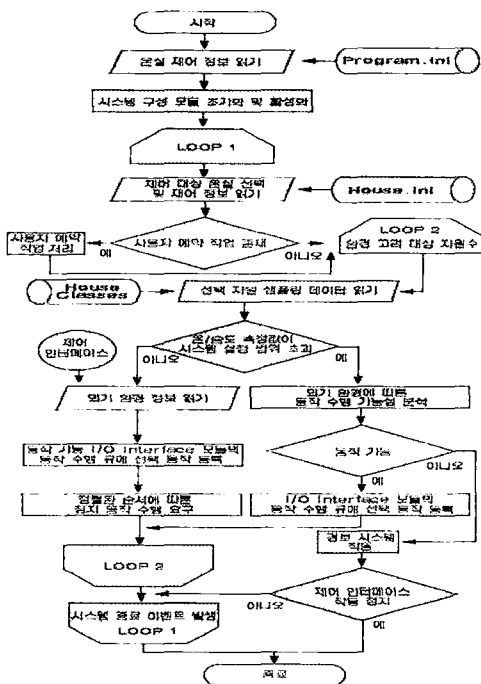


그림 4. 시스템의 총괄 작업 흐름도
Fig. 4. Global flowchart of system

그리고 완전히 독립된 클래스의 설계에 따른 시스템 확장성과 병행성을 제공하여 처리의 속도를 향상시켰다. 그림 5는 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.

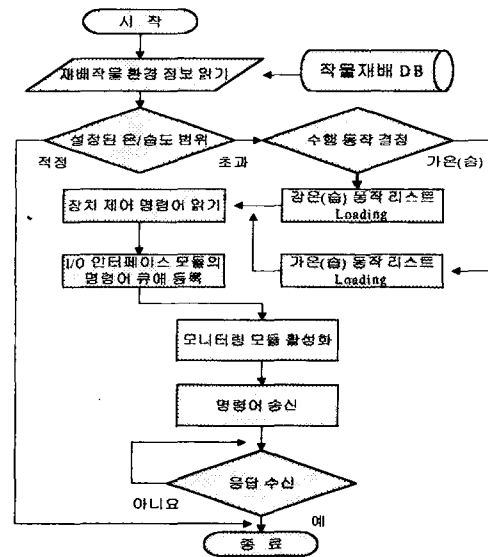


그림 5. 온·습도 처리 흐름도
Fig. 5. Process flowchart of temperature and humidity

기존의 논문[7], [11]에서는 온실 환경 제어를 위한 판단 요소로서 내기 환경 요소로 측정 범위를 국한시켰다. 그래서 기상 재해나 우천시 분석 환경 요소의 부재에 따른 판단 오류에 기인되어 환경 제어시 시스템 신뢰도와 전체 온실 자원의 위험도가 매우 높았다. 이에 온실 환경 제어 정보 시스템은 환경 제어를 위한 단위 동작 판단을 위해 외기 환경을 측정하여 시스템 환경 제어에 반영한다. 외기 환경 측정은 내기 환경 측정과 같이 I/O Interface Module과 컨트롤러와 RS-232C를 통한 정의된 명령어 테이블의 명령어 패치와 송·수신을 통해 수행된다. 그림 6은 측정된 외기 환경에 따른 처리 과정을 나타낸다.

온실 자원의 현재 상태를 감시하고 문제 발생 시 적절한 대응을 수행하기 위해 온실 제어 정보 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 직렬 통신을 수행하기 위해 우선 순위를 가지는 큐

를 이용한다.

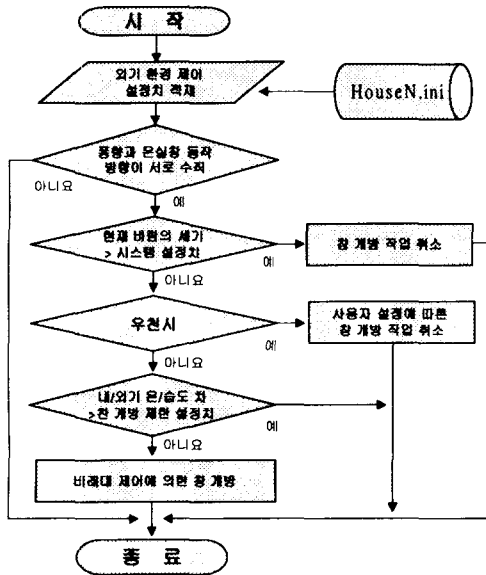


그림 6. 외기 환경 처리 흐름도
Fig. 6. Process flowchart of the open air

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 종료되는지를 확인하여 동작 결과에 따라 정보 시스템과 연동하여 감시를 종료한다. 그림 7은 모니터링 모듈의 제어 처리 흐름이다.

복구 불가능한 제어 상태에서부터 신속한 회복을 위해 정보 시스템의 효과적 제어는 필수적이다 [1],[3]. 온실 환경 제어 정보 시스템의 정보 시스템은 사용자의 위치와 시간 제약적인 환경을 고려해서 개발하였다. 즉, 정보 시스템은 사용자의 요구나 시스템 디폴트 설정을 통해 3가지 레벨 중 하나로 설정된다. 레벨의 증가에 따라 사용자의 효과적 정보 메시지 인지를 위해 메시지의 전달 방법과 형태가 달라진다.

각 레벨의 기능으로 레벨 1은 텍스트 정보 메시지를 화면에 표시하며, 레벨 2는 레벨 1의 정보 메시지에 추가하여 온실 내·외부의 스피커를 통

해 정보 메시지를 음성으로 전달한다. 레벨 3은 레벨 2에 추가하여 정보 메시지를 휴대폰을 통해 시스템에 설정된 메시지 수신자에게 정보 메시지를 전달하게 된다. 정보 시스템의 다양한 메시지 전달 형태를 통해 기존의 정보 시스템의 공간 제약적인 단점과 정보 메시지에 대한 사용자 인지 시간을 단축시킬 수 있었다.

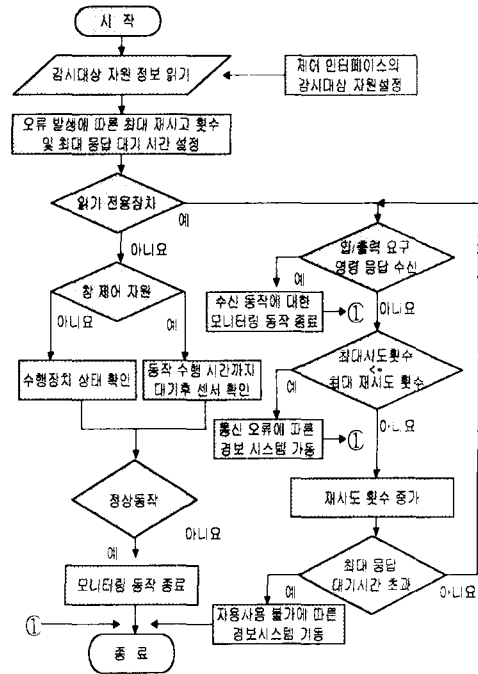


그림 7. 모니터링 모듈의 제어 흐름도
Fig. 7. Control flowchart of the monitoring module

그림 8은 오류 발생시 수행되는 정보 시스템의 처리 흐름도이다. 휴대폰을 이용한 정보 메시지 전달을 위해 무선 문자 메시지(Short Message Service) 서비스를 이용하였다.

SMS 서비스는 디지털 이동전화의 액정화면(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.

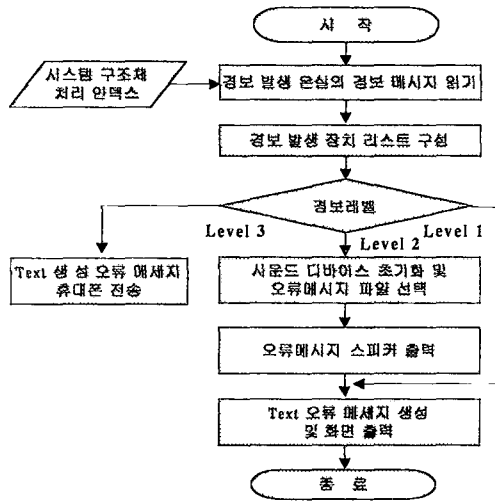


그림 8. 경보 시스템 처리 흐름도
Fig. 8. Process flowchart of alarm system

4. 데이터베이스의 설계

데이터베이스는 표 2, 표 3에 정의된 테이블 명세에 따라 Paradox를 이용하여 설계하였다. BDE 엔진은 현재 온실 장치의 상태값을 DB에 저장하기 위해 동적 생성된 SQL문을 처리한다. BDE 엔진 내부의 TDataSet 클래스는 테이블을 직접 조작하는데 필요한 메소드를 가진 추상 클래스이다. TDBDataSet 클래스는 패스워드와 특정 테이블에 대한 연결을 수행하며, TTable은 물리적 재배 작물 DB의 색인과 일대다 관계의 두 테이블 연결 정보를 가지고 있다. TQuery는 환경 정보 DB의 자료 처리를 위한 SQL 문을 가지고 있다. 자료 처리 과정의 흐름도는 그림 9와 같다.

기존의 시스템에서 작물 재배를 위한 사용자 제공 정보에 대한 부분적인 DB가 구축되어 있다. 그러나 재배 작물의 영역별 분류와 GUI의 비효율적인 구성으로 인해 사용자에 대한 운영상의 편리성이 고려되어 있지 못하고 있다.

본 논문에서는 이러한 점을 개선하여 체계화된 DB 정보 제공[2],[5]을 위해 환경 정보 DB와 재배 작물 DB로 구분하여 설계하였다. 물리적 DB 테이블의 분리를 통해 검색시 불필요한 정보의

메모리 적재를 최소화하여 검색 속도를 향상시켰다. 그리고 사용자 서비스를 위한 시스템 작업 처리시 테이블 특성에 따른 전문적 기능을 제공할 수 있다.

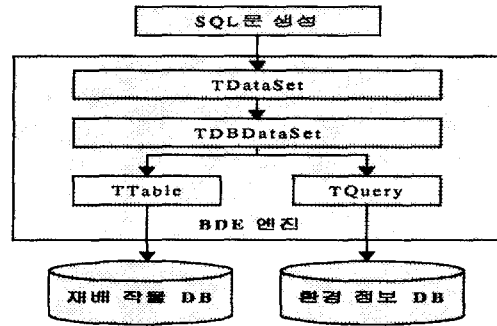


그림 9. BDE를 이용한 DB 처리 흐름도
Fig. 9. Process flowchart using BDE

재배 작물 DB는 재배 작물의 종별 구분이 명확히 이루어지도록 설계하였으며, 사용자의 효과적 작물 정보 탐색을 위해 기능별 테이블 항목 구성으로 필드별 패턴 검색과 네비게이터가 효과적으로 수행되도록 구성하였다. 그리고 작물 재배 환경 정보를 추가하여 작물 재배 환경 설정시 사용자의 추가적인 설정없이 자동 환경 설정을 하기 위한 필드가 추가되었다. 표 2는 온실 환경 제어 정보 시스템의 재배 작물 DB 구축을 위해 설계된 테이블을 나타내고 있다.

시스템 운영시 온실의 내·외기 현황을 파악하여 시스템의 현재 상태와 앞으로의 운영을 결정하는데 환경 정보 DB의 히스토리 정보 제공은 필수적이다. 이러한 환경 정보 제공을 위한 DB 테이블은 빈번히 이루어지는 자료 누적 연산의 효율을 높이기 위해 각 기능 요소에 대한 수치 표현의 정밀도와 처리 속도를 고려하여 설계해야 한다. 추가적으로 샘플링 자료의 수집 시간의 정확한 명시를 통해 자료 통계시 적절히 처리될 수 있도록 한다. 표 3은 환경 정보 DB 테이블을 나타내고 있다.

표 2. 재배 작물 DB 테이블
Table 2. The table of corps DB

| 필드명 | 타입 | 내용 |
|------------|-----------|---------------|
| Name | char(255) | 작물명 |
| EName | char(255) | 학명(영문) |
| Source | char(255) | 원산지 |
| Trade | char(255) | 대표종 |
| AverDat | char(255) | 통계자료 파일 경로 |
| Property | char(255) | 작물 특성 요약 |
| GlowDat | char(255) | 재배작물 파일 경로 |
| ImpagePath | char(255) | 작물 이미지 파일 경로 |
| Category | char(255) | 작물 구분 |
| MinTemp | int | 작물 재배 최저 온도 |
| MaxTemp | int | 작물 재배 최저 온도 |
| MinHud | int | 작물 재배 최저 습도 |
| MaxHud | int | 작물 재배 최저 습도 |
| MinSun | int | 작물 재배 최저 일사량 |
| MaxSun | int | 작물 재배 최저 일사량 |
| MinCO2 | int | 작물 재배 최저 CO2량 |
| MaxCO2 | int | 작물 재배 최저 CO2량 |

표 3. 환경 정보 DB 테이블
Table 3. The table of environmental information DB

| 필드명 | 타입 | 내용 |
|---------|------------|-----------|
| Years | (yy/mm/dd) | 년월일 |
| Times | (hh/mm/ss) | 시분초 |
| Temp1 | f(6,3) | 온도계 1 측정값 |
| Temp2 | f(6,3) | 온도계 2 측정값 |
| Hud1 | f(6,3) | 습도계 1 측정값 |
| Hud2 | f(6,3) | 습도계 2 측정값 |
| WindDir | f(6,3) | 풍향 측정값 |
| Water | f(6,3) | 관수 측정값 |
| SunFall | f(6,3) | 일사량 측정값 |
| CO2 | f(6,3) | 이산화탄소 측정값 |

III. 구현

1. 개발 환경

시스템 개발은 개인용 컴퓨터, 멀티미디어 재생 장치와 시스템 제어를 위한 제어 인터페이스 장치로 구성되며, 소프트웨어는 DB 구축을 위한 InterBase 5.0, 시스템 구축을 위한 BC++3.0, 멀티미디어 데이터 편집 도구, 그리고 Windows

95/98/NT 환경에서 동작 가능하도록 개발하였다.

본 시스템을 구성하는 각 요소들은 기능 중심으로 설계된 클래스로 구성되어 있다. 시스템 제공 기능은 특성에 따라 온실 상태 제어와 작물 정보 조회로 구분된다. 그림 10은 온실 제어 정보 시스템에 제공하는 시스템 메뉴 구성도이다. 메뉴는 계층적으로 구성되어 있으며, 상위 메뉴에 포함된 하위 메뉴는 해당 영역의 세부적인 기능을 제공한다.

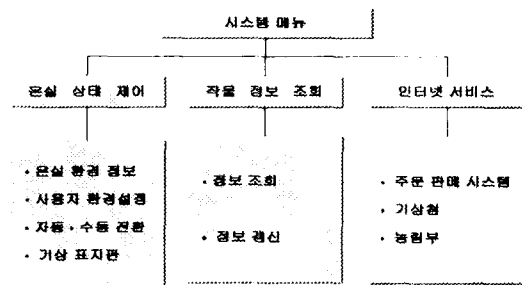


그림 10. 시스템 메뉴 구성도
Fig. 10. Configuration of system menu

전체 시스템의 클래스 모듈들은 특성에 따라 네가지 형태로 구분할 수 있다. 모니터링을 위한 다동 온실 클래스들과 컨트롤러를 통제하기 위한 제어 클래스와 구조체, 모니터링과 자동 환경 제어를 위한 시스템 제어 엔진 클래스, 그리고 사용자에게 유용한 서비스를 제공하기 위한 응용 클래스들로 구분된다.

각 클래스들은 구분된 특성에 따라 상호 유기적으로 결합되어 작업을 처리한다. 또한, 온실의 다동 제어를 위한 온실 클래스의 계층적 구조를 통해 일관성 있는 접근 제어를 제공한다. 각 온실 클래스는 온실 자원을 제어하고 현재 상태를 저장하기 위한 멤버 클래스와 변수, 메소드들을 포함하고 있다. 온실 클래스에 포함된 온실 자원들도 클래스화 되어 있기 때문에 실행시간에 커스터마이징되는 것이 가능하다. 그림 11은 온실 클래스가 가지는 멤버 클래스, 변수, 메소드들 중 일부를 나타내고 있다. 온실의 자동 환경 제어 모니터링 시스템은 컨트롤러를 통제하기 위해 제어 클래스 사용한다. 제어 클래스는 동작 대상 장치

를 표현하고 수행을 위한 동작 명령어, 장치 상태 값 등을 포함하고 있다. 제어 클래스는 온실 자원의 특성에 따라 ReadOnly 자원 클래스와 Read/Write 자원 클래스로 구분된 멤버 클래스들을 가진다. ReadOnly 자원 클래스는 주로 센서의 측정값과 제어명령어, ID 등을 가지며, 같은 형태의 자료 구조를 가지고 있기 때문에 확장이 용이하다.

```

Class House
{private:
    //해당 온실 초기화
    void __fastcall initial();
    //온실 창 개방/폐쇄 데드라인 읽기
    void __fastcall SaveDeadLine(int HouseNum);
    //Context Switch Loading
    void __fastcall LoadHouseInf(int HouseIndex);
    //Context Switch Saving
    void __fastcall SaveHouseInf(int HouseIndex);
public:
    // User declarations
    //ReadOnly Type Class Declearation
    READDEV_ ReadDev[MAXDEV];
    //Read/Write Type Class Declearation
    Window_ Window; //윈도우 클래스 선언
    CoolingFan_ CoolingFan; //쿨링팬 클래스 선언
    Power_ Power; //동력 클래스 선언
    Boiler_ Boiler; //난방기 클래스 선언
    Curtain_ Curtain; //커튼 클래스 선언
    WaterSup_ WaterSup; //관수 공급 클래스
    Co2_ Co2; //Co2 클래스
    int OpenRateEnable[8]; //비례대 제어 );
    
```

그림 11. 온실 클래스의 구성 멤버
Fig. 11. Member of greenhouse class

그러나 Read/Write 자원 클래스는 주로 온실 모터나 특정 장치를 위한 자료 구조를 가지기 때문에 독립적으로 설계되어 있다. 제어 클래스의 컨트롤러의 제어를 위한 처리 구성도를 그림 12에서 나타내고 있으며, 그림 13은 제어 클래스를 구성하는 장치 클래스를 나타낸다.

컨트롤러는 통신 상에서 센서 읽기 명령 수행 후 결과 프레임 수신시에 오류가 발생할 수 있다. 이러한 경우 통신 상의 무한 대기 상태가 발생할 가능성이 있다.

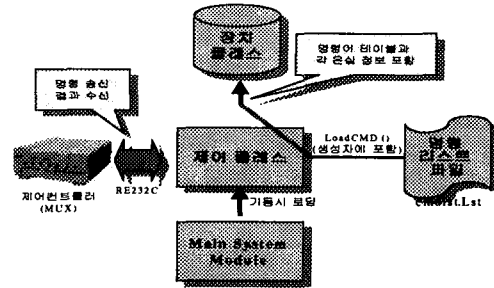


그림 12. 제어 클래스의 처리 구성도
Fig. 12. Process configuration of control class

```

class RW_DEV_ //Read/Write 장치 클래스
{public:
    int Index; //1. 인덱스
    AnsiString Dev; //2. 장치명
    AnsiString CMD; //3. 동작명령어
    char InAction; //4. 정방향 동작
    char ReAction; //5. 역방향 동작
    char StopAction; //6. 정지 동작
    int State; //7. 장치상태값
    int Limit; //8. 리미터
    RW_DEV_(); };
class READ_DEV_
{public:
    int Index; //1. 인덱스
    AnsiString Dev; //2. 장치명
    AnsiString CMD; //3. 동작명령어
    int Dat; //숫자 데이터
    char sDat[9]; //문자열 데이터
    READ_DEV_(); };
    
```

그림 13. 제어 클래스의 장치 멤버 클래스
Fig. 13. Devices member class of control class

그리고 제어기의 통신 선로가 명령 프레임의 순차적 처리 특성에 기인되어 각 제어 명령어의 수행을 위해서 우선 순위 스케줄이 필요하다. 이러한 문제를 보완하기 위해 온실 환경 제어 정보 시스템의 제어 클래스는 우선 순위를 가지는 큐를 이용하여 명령을 송·수신한다. 우선 순위는 일반적인 경우 선입선출에 따른 우선 순위가 할당되며, 긴급 수행 명령어는 미리 정의된 장치 우선 순위에 따라 명령어 발생시 초기 설정 우선 순위가 높게 할당되어 큐에 삽입된다. 큐는 매 실

행시 우선 순위에 따라 명령어를 재정렬한 후, 명령을 수행한다.

그리고 송신 명령에 대한 응답이 제한 대기 시간을 초과한다면, 사용자 정의나 시스템 디폴트(3회)에 의해 설정된 횟수만큼 송신 재시도를 수행하며, 재시도 횟수를 초과하였을 때 경보 시스템을 기동한다. 그림 14는 명령어 큐의 클래스 구조를 나타내고 있다.

```
class CMD_Q // 명령어 저장큐
{private:
    //명령어 저장 배열
    AnsiString CMD[MAXELEMENT];
    //우선순위 저장 배열
    int Pri[MAXELEMENT];
public:
    int pCMD; //삽입 인덱스
    int iCMD; //읽기 인덱스
    CMD_Q(); //생성자
    //명령어 삽입 메소드
    int setCMD(AnsiString sCmd);
    AnsiString getCMD(); //명령어 읽기 메소드
    AnsiString View(int i); //i번째 CMD 내용 출력);
```

그림 14. 명령어 큐의 클래스
Fig. 14. Class of command queue

자동화 제어 엔진은 온실 환경 제어 정보 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 4는 이러한 동작 리스트와 우선 순위를 나타낸다.

그림 15의 제어 엔진 클래스는 다중 온실 제어를 위해 시스템 독립적 클래스로 설계되었다. 엔진은 각 온실에 대해 매핑되며, 할당된 대상 온실 자원의 상태를 저장하기 위한 멤버 클래스와 메소드로 구성된다.

표 4. 엔진 제어 동작 리스트와 우선 순위
Table 4. Engine control action list and priority

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

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

그림 15. 환경 제어 엔진 클래스
Fig. 15. Engine class for environmental control

환경 자동 제어시 Main System Module은 각 온실을 제어하기 위해 엔진을 순차적으로 할당한다

다. 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.

2. 구현 결과

본 연구에서 구현한 온실 제어 정보 시스템은 Windows 95/98, Windows NT에서 시분할 이벤트 복합 방식으로 단동 및 다동 온실에 대한 자원을 모니터링한다. 그리고 사용자 서비스를 위한 환경 정보를 DB로 구성하며, 오류 발생시 경보 레벨에 따른 메시지를 사용자에게 전달한다. 추가적으로 사용자에게 유용한 웹 링크 서비스도 제공한다.

가) 온실 환경 제어 모니터링

시스템을 기동시키면 로고 수행과 동시에 그림 16의 I/O Interface Module의 제어 윈도우가 표시되는데, 수행 명령의 IRAF는 외기 온·습도를 체크하는 명령이며 현재 온실의 각 자원에 대한 상황 데이터가 실시간으로 표시된다.

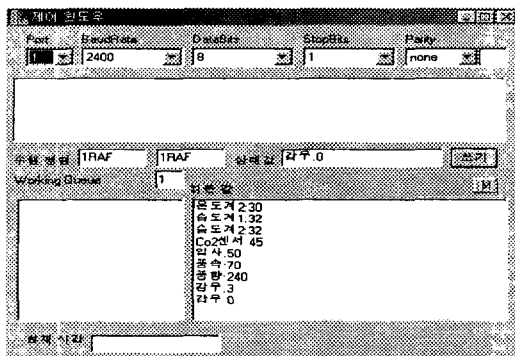


그림 16. I/O Interface module의 제어 윈도우
Fig. 16. Control window of I/O interface module

I/O Interface Module은 컨트롤러와의 통신을 통해 현재 온실 자원의 초기 상태값을 수신받는다. 그리고 사용자의 편의성을 향상시키기 위해 터치 스크린을 이용하여 모니터링 윈도우의 왼쪽 창의 각 장치를 터치할 경우와 오른쪽 창의 각

장치에 대한 버튼을 이용하여 동작 수행이 양쪽으로 가능하도록 하였다. 다음은 모니터링시 주로 사용되는 주메뉴를 기술한다.

● 온실 환경 정보 메뉴

온실 내·외부 자원 모니터링 상황을 나타내는 온실 환경 정보 메뉴를 선택하면 그림 17과 같이 표시된다. 모니터링을 위한 동작은 사용자의 편의를 위하여 각 장치를 터치하는 것과 오른쪽 버튼을 이용하여 각 장치의 동작을 병행하여 수행할 수 있도록 구현하였다. 또한, 위급한 환경의 변화에 따라 각 장치를 긴급히 정지시켜야 할 경우는 총괄 시스템의 “정지” 버튼을 이용하여 동작을 정지시킬 수 있다. 각 장치가 동작 중일 때는 각 자원들이 점멸하게 되는데, 특히 축창과 천창의 개폐시 검은 점(모터)이 점멸하면서 동작하게 된다.

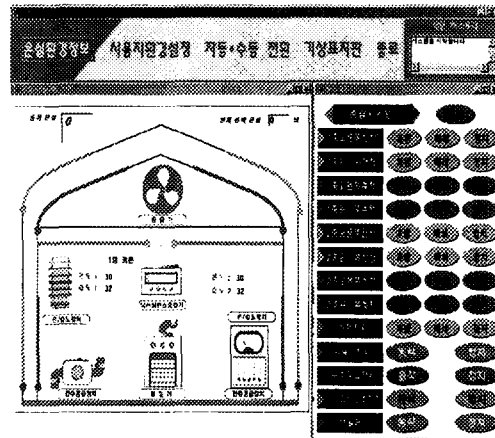


그림 17. 주메뉴와 모니터링 윈도우
Fig. 17. Main menu and monitoring window

● 수동·자동 전환 메뉴

사용자 요구에 따라 수동·자동 운영 모드를 제공한다. 수동 운영 모드에서는 온실의 각 자원에 대해 동작 윈도우를 사용해서 통제 가능하다. 자동 운영 모드는 각 대상 온실의 작물 재배를 시스템 제어 엔진을 이용해 자동화시키는 운영 모드이다.

● 사용자 환경 설정 메뉴

각 온실의 자원을 사용자가 실행 시간에 동적으로 커스터마이징하기 위한 기능 제공을 위한 환경 설정 윈도우를 제공한다. 환경 설정 윈도우는 각 자원의 사용 유무와 제어 설정치를 온실 제어 정보 시스템 레지스트리에 기재하여 실행시 이를 반영한다.

● 기상 표시판 메뉴

온실 내부의 작물 재배 환경은 온실 외부의 기상 환경에 밀접한 관계를 가진다. 온실 환경 제어 정보 시스템은 최적화된 작물 재배 환경 조성을 위해 외부 기상 상태를 점검하여 환경 제어에 반영한다. 외부 기상 상태 확인은 일사 센서, 감우 센서, 강우량 센서, 풍향 및 풍속 센서에 의해 현재 기상 상태에 대한 정보를 제공한다.

나) 작물 재배 정보 DB 검색

4절의 데이터베이스의 설계를 구현한 후, 작물 재배 정보, 작물별 설정, 데이터 통계 등의 검색 결과를 표시해 준다.

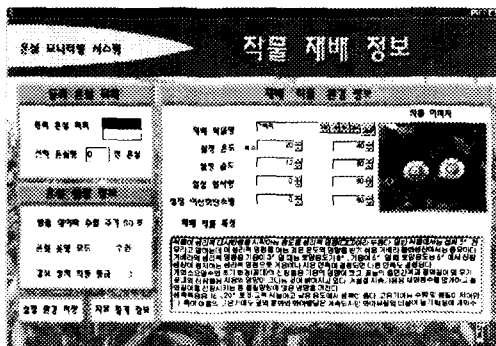


그림 18. 작물 재배 정보 윈도우
Fig 18. Window for corps information

● 작물 재배 정보 윈도우

사용자의 온실 작물 재배 환경 설정을 위한 작물 재배 윈도우는 그림 18과 같다. 사용자는 재배하고자 하는 대상 온실을 선택한 후, 재배 작물명을 선택한다. 사용자의 선택에 대해 시스템은 선택된 대상 온실의 재배 작물에 대한 환경 정보를

DB에서 조회하여 자동 설정하며, 자동 설정된 작물별의 내용을 사용자가 변경할 수도 있다. 변경된 내용은 환경 정보 DB에 업데이트된다.

● 재배 작물 DB 검색 윈도우

재배 작물에 대한 정보 제공을 위해 환경 제어 시스템은 작물 정보 DB를 가진다. 등록 작물 목록에는 화훼와 채소로 나뉘어지고, 작물 정보 보기와 작물 정보 검색 그룹으로 구분된다. 그림 19는 작물 정보 검색을 위한 윈도우이다.

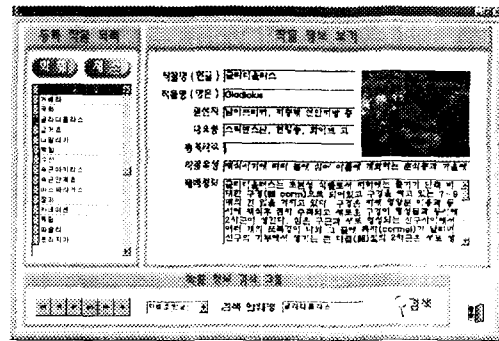


그림 19. 작물 정보 검색 윈도우
Fig. 19. Window for corps information search

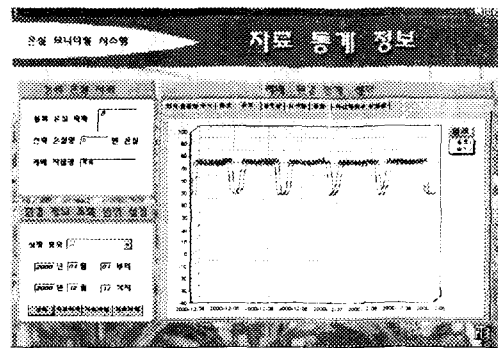


그림 20. 자료 통계 윈도우
Fig. 20. Window for data statistics

● 자료 통계 윈도우

작물 재배 정보 DB를 통하여, 각 단위 온실에 대한 온·습도, 일사량, 강우량, CO₂ 발생량 등의

자료 통계 정보를 그림 20과 같이 표시한다. 즉 온실 내 각 자원의 샘플링 수치를 10분간격으로 텍스트나 그래프 형태로 표시할 수 있고, 화면 왼쪽 아래의 설정 범위를 지정한 만큼의 데이터를 삭제, 백업, 저장 등도 가능하다.

IV. 기존 시스템과의 비교 및 분석

본 논문에서 개발된 자동 환경 제어 모니터링 시스템은 기존 연구에서 구현된 시뮬레이션 환경에서의 제어를 현장의 온실에 직접 적용하여 사용자 신뢰도를 향상시켰다. 기존 시스템의 제한된 자연 환경 조건과 실제 하드웨어 제어 동작에 대한 추상적 표현으로 인한 부적절함을 보완하였고, DOS 환경하에서의 모니터링 시스템과 on/off식의 제어반만을 이용한 시스템에 비해 효율적이며, 여러 동의 온실을 제어하는 편리성을 제공하고 있다.

기존의 시스템에서는 원격지에서 오류 발생 상태를 알 수 없고, 한개의 on/off식 제어반으로는 여러 동의 온실을 제어할 수 없다. 그리고, 온실 환경 변화에 따른 장기간의 데이터를 보관할 수 없으며, 주기적 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 20% 이상(50동까지 확장가능) 증가시킬 수 있다. 온·습도 등 매일의 데이터를 정확하게 저장함으로써 재배 기술의 향상과 축적을 시킬 수 있다. 그리고 기기 자동화로 인해 관리자가 온실 지역을 벗어나도 핸드폰으로 온실의 오류 발생 메시지를 전달해 주기 때문에 사용자의 편의성을 높였다. 그러나, 컴퓨터 제어를 위한 컨트롤러와 PC 사용에 따른 전체 시스템 구축 비용이 고가인 단점이 있다. 표 5는 기존 연구의 시스템과의 기능을 비교한다.

표 5. 기존 시스템과의 비교
Table. 5. Comparison for existing system

| 구분 | 기존 시스템 | 본 논문의 시스템 |
|---------------------|--------|-----------|
| 온실 제어 | 가능 | 가능 |
| 데이터 장기 보관 및 업데이터 기능 | 불가능 | 가능 |
| 오류 상태 파악 및 경고 기능 | 불가능 | 가능 |
| 편의성 | 저 | 고 |
| 실시간 상황 감시 | 불가능 | 가능 |
| 시스템 가격 | 저가 | 고가 |
| DB 관리 기능 | 불가능 | 가능 |

V. 결론

본 논문에서 개발한 실시간 자동 환경 제어 모니터링 시스템의 기법은 실제 현장에 적합하고 어떠한 선택 사양에도 편리하게 적용될 수 있는 시스템으로서 시뮬레이션 수행시 발생하는 동작의 추상성과 제한된 시뮬레이션 환경 조건을 실제 현장에서 적용한 결과를 비교 분석하여 보완하였다. 그리고 제어 수행시 발생하는 실제 장치 오류와 각 상태에 대한 모니터링을 위한 다음의 동작 판단은 내·외기 환경 정보 등 온실의 자원에 대한 상황 데이터가 실시간으로 표시되므로, 시스템의 안전성을 크게 높였고, 온실의 상황 감시가 효율적이다.

특히, 본 시스템은 양식장 자동화, 공장 자동화 등의 자동화 연구 분야에 효과적인 응용이 될 것이다. 향후 인터넷을 이용한 실시간 환경 모니터링과 원격 제어를 위한 시스템의 개발로 지역의 균형 발전을 위해 농가에서도 인터넷을 이용한 정보를 획득함으로써 첨단 과학적 영농을 도모하고자 한다.

참고문헌

[1] Alan Burns, Andy Wellings, "Real-Time Systems and Programming Languages", ADDISON -WESLEY, pp.95-125. 1996.

- [2] W. Lu, J. Han, and B. C. Ooi, "Discovery of General Knowledge in Large Spatial Database" , Proc. Far East Workshop on Geographic Information Systems, Singapore, June 1993, pp. 275-289.
- [3] 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.
- [4] Keith A. Butler, Robert Jacob, Bonnie E. John, "Introduction and Overview of Human- Computer Interaction", CHIVAS, April 1994.
- [5] M. Chen, J. Han, and P. Yu, "Data Mining: An Overview from Database Perspective", IEEE Transactions on Knowledge and Data Eng., December 1996, pp. 866-883.
- [6] 조성연, 이병래, 백광진, 김태윤, "클라이언트/서버에 기반한 공장 자동화 시스템의 설계와 구현", 정보처리학회 발표, pp.851-854. 1998.
- [7] 권영삼, "국내 원예시설의 유형별 특성과 발전 방향", 국내외 원예 시설의 특성과 시스템에 관한 심포지엄, 한국시설원예연구회, pp.111-125, 1995.
- [8] 민영봉, 박중훈, 이상옥, 정대상, "온실의 복합환경제어 시스템의 구성", 경상대 시설 원예 연구, pp.151-165, 1994.
- [9] 김문기, "마이크로 컴퓨터를 이용한 온실 환경 제어", 경상대학교 시설 원예 연구, pp.261-331, 1994.
- [10] 조성연, 이병래, 백광진, 김태윤, "클라이언트/서버에 기반한 공장 자동화 시스템의 설계와 구현", 1998년 한국정보처리학회 추계 학술발표논문집 제 5권 제 2호, p851-854.
- [11] 홍순호, "Automatic Control of Growth Environment for Plant Factory", 서울대학교 박사학위 논문, 1995. 1
- [12] 박홍복, "시설 원예를 위한 비닐 온실 자동화 관리 시스템 개발", '99년도 부경대학교 산학연 컨소시엄, 1998. 10.
- [13] 서상진, 박홍식, 박홍복, "온실 환경 자동화

시스템의 설계 및 구현", 한국정보처리학회 제12회 추계학술대회, paper-463, 1999. 10



김 대 업(Dae-Up Kim)
2000년 동서대학교 컴퓨터공학(공학사)
2000년~현재 부경대학교 대학원 전자계산학과 석사과정



엄 현 서(Hyun-Seo Eom)
2000년 부경대학교 전자계산학과 (이학사)
2000년~현재 부경대학교 교육대학원 전산교육전공 재학 중



정 광 하(Kwang-Ha Jung)
1999년 동아대학교 원예학과 (농학사)
2000년~현재 부경대학교 산업대학원 전산정보학과 재학중



박 흥 복(Hung-Bog Park)
1982년 경북대학교 컴퓨터공학 (공학사)
1984년 경북대학교 대학원 컴퓨터공학 (공학석사)
1995년 인하대학교 대학원 전자계

산학전공(이학박사)
1984년~1995년 동명대학 전자계산학과 부교수
1996년~현재 부경대학교 전자계산학과 부교수
※관심분야 : 실시간 시스템, 멀티미디어 응용, 원격 제어, 컴파일러 및 객체지향 프로그래밍