

최 종  
연구보고서

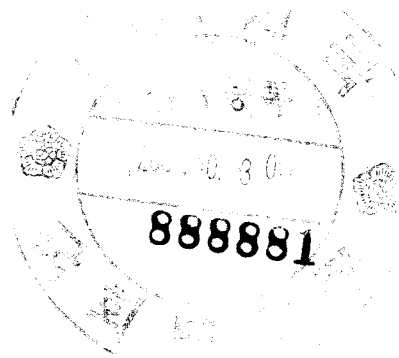
원예작물의 저온저장 중 환경관리

자동화 시스템 개발

Automatic Monitoring System for Cool Stores  
in Postharvest Horticultural Products

연구기관  
상명대학교

농림부



# 제 출 문

농림부 장관 귀하

본 보고서를 “원예작물의 저온저장 중 환경관리 자동화 시스템 개발”의 최종보고서로 제출합니다.

2001. 10. 11.

주관연구기관명 : 상명대학교

총괄연구책임자 : 양 용 준

세부연구책임자 : 양 용 준

연 구 원 : 이 경 아

연 구 원 : 김 경 민

연 구 원 : 김 윤 희

세부연구책임자 : 박 병 수

연 구 원 : 유 희 정

연 구 원 : 윤 희 윤

연 구 원 : 전 소 영

협동연구기관명 : 한양대학교

협동연구책임자 : 최 명 렬

연 구 원 : 이 상 민

연 구 원 : 권 재 우

연 구 원 : 박 재 현

연 구 원 : 황 희 찬

연 구 원 : 남 상 섭

연 구 원 : 김 중 목

# 여 백

# 요 약 문

## I. 제목

원예작물의 저온저장 중 환경관리 자동화 시스템 개발

## II. 연구개발의 목적 및 중요성

원예산물의 수확 후 저장 및 유통과정에서 발생하는 손실은 우리 나라 총 생산량의 30-40% 정도로 추산되며 저장고가 충분히 설치되지 않고 수확 후 관리 유통관행상 이런 손실율을 줄이기는 쉽지 않다. 현재 농가의 저온저장고의 면적이 증가하는 추세임에도 불구하고 작물의 저온저장에 따른 환경조성과 저장요인의 변화를 수시로 관리, 예측하지 못하는데 따른 피해가 해마다 증가하고 있다. 대형 저장고의 경우 단일 저장고 내 수십 톤에서 수백 톤이 저장되는 현실인데 이 저장고 내 온도의 심한 편차 (최소 2-3°C 정도) 때문에 실제로 관리자의 입장에서 적정 온도의 유지란 큰 부담이 되므로 많은 경우 비교적 온도를 높게 유지하는 경우가 관행으로 되어 있다. 따라서 이 경우 정확한 온도센서를 배치한 뒤 컴퓨터에서 자동 관리하게 된다면 저온장해로 인한 손실은 크게 예방하게 되리라 본다.

저온저장고내 환경중 온도, 습도 다음으로 산소의 부족과 과도한 이산화탄소의 집적에 따른 피해가 과수 및 채소에서는 우려되는 상황이다. 이것은 작물에 따른 적정 산소 및 이산화탄소 농도에 대한 정보가 있어야 관리가 가능하지만 실제 저온 저장시 이 가스에 대한 인식은 농가에 거의 알려져 있지 않은 실정이다. 수확 후 발견되는 생리적 장해현상으로 숙성의 빠른 진행이나 노화 그리고 저장고내 이산화탄소의 집적이나 산소의 결핍으로 해석되는 내부갈변 현상 등이 대표적인데 이런 장해발생을 조기 예측하기 위하여 환경관리 자동화 시스템의 개발은 매우 중요하다. 저온저장고 내 환경, 즉 온도, 상대습도, 이산화탄소 그리고 산소에 대한 농도변화에 대한 정보를 컴퓨터에 전송한 뒤 자료를 축적해 함으로써 이러한 환경정보를 관리할 수 있는 자동화시스템의 개발은 다음과 같은 여러 가지 부가가치를 제공하여 줄 수 있을 것이다; 1) 최적 온도, 습도, 이산화탄소, 산소 등의 저장 환경인자를 최적 조절 상태로 유지시킬 수 있고, 2) 작물에 따른 특

이한 수확후 생리적 숙성 및 노화과정에 대한 변화를 용이하게 예측하게 하고, 3) 부패 미생물에 의한 피해를 최소화하며, 4) 온도 및 습도 유지에 관한 정보를 관리사무실이나 집에서 쉽게 관찰할 수 있는 관리상 편리성을 제공할 수 있을 것이다.

본 연구에서는 원예작물의 저장고 내 품질관정을 목적으로 이와 관련된 여러 환경요인들을 일차적으로 센서에서 감지하고 그 정보를 컴퓨터에 자동 입력하여 이러한 환경을 자동으로 관리할 수 있는 시스템의 개발을 목적으로 하고 있다.

### III. 연구개발내용 및 범위

#### 1. 원예작물의 수확 후 생리특성 조사

- 1) 원예작물의 품목차이에 따른 수확후 생리 및 적용성 여부 조사
- 2) 작물 및 저장온도에 따른 생리특성 연구
- 3) 작물에 따른 저장성 판단 모델정립

#### 2. 원예작물의 환경관리 자동 측정 시스템의 구축 및 정보 측정

- 1) 자동 측정 시스템의 구축 및 측정
- 2) 저장고의 설정기준치와 자동측정 시스템간의 오차범위 최소화
- 3) 개선된 방식의 자동 측정 시스템 개발

#### 3. 상업용 저장고의 규모에 따른 환경관리의 통일된 시스템을 개발

### IV. 연구개발 결과 및 활용에 대한 건의

#### 1. 원예작물의 수확 후 생리특성 조사

저장 온도에 따른 후지 사과의 호흡량을 조사한 결과 전형적인 호흡급등형 과일 (climacteric type)의 양상을 나타내었다. 4℃와 10℃ 처리구에서는 저장 15일까지 호흡이 증가하여 저장 15일째 호흡 최고치를 나타내다가 이후 감소하여 저장 45일째부터는 호흡량에 큰 변화를 보이지 않았다. 전체적인 에틸렌 및 에탄올 생성량은 저장 온도가 높을수록 크게 나타났다. 신고배의 호흡량 및 에틸렌 발생 조사 결과 비호흡급등형 과일 (non-climacteric type)의 양상을 나타내었다.

## 2. 원예작물의 환경관리 자동 측정 시스템의 구축 및 정보 측정

### 1) 자동 측정 시스템의 구축 및 측정

작물이 들어있는 여러 개의 저장용기 내부의 온도, 습도 및 이산화탄소, 산소, 에틸렌 농도 측정을 위해서 해당하는 각각의 센서를 이용, 환경정보를 용기별로 순차적으로 자동 측정하는 방식이다. 이때 저장용기 내의 가스샘플을 수집하기 위해서 모터를 설치하고, 모터에 의해 저장 용기내의 가스를 일정량 흡입하게 된다. 이때 얻어진 가스샘플은 저장 용기와 연결된 흡입관을 통해 각 센서까지 전달된다. 온, 습도 센서를 비롯한 가스 측정 센서는 종류별로 각 1개씩이 사용되었다. 이때 각 센서를 통해 나온 전류 (혹은 전압) data는 호스트 컴퓨터상의 응용 소프트웨어상에서 사용자가 원하는 단위인 % (습도, 이산화탄소, 산소)나 ppm(에틸렌), 혹은 ℃(온도)로 변환되며, 이 모든 과정은 컴퓨터 프로그램상에서 제어가 가능하다

### 2) 개선된 방식의 자동 측정 시스템 개발

모터에 의한 가스 흡입방식의 자동측정 시스템은 모터에 의해 흡입되는 정확한 공기의 양을 알기 어렵고 또한 이에 의한 저장고내 기체 환경 조성의 변화가 발생하기 때문에 정확한 가스농도를 측정하기 어려운 문제점이 나타났다. 따라서 모터를 이용하여 흡입한 저장고내 가스를 저장고 밖에 위치한 가스 센서까지 운반하여 감지하는 형식이 아닌 가스센서를 직접 저장고내에 위치시켜 농도를 감지하는 방식으로 개선하였다.

### 3) 상업용 저장고의 규모에 따른 환경관리의 통일된 시스템을 개발

지금까지 하드웨어 및 소프트웨어 개발팀에 의해 공동으로 구축된 실험실 차원의 자동 측정 시스템 장치를 상업용 저장고 규모로 확대, 적용하기 위하여 2 톤 규모 상업용 pilot 저장고 2개에 각각 온도, 습도, 이산화탄소, 산소, 에틸렌 센서를 배치하고 사과와 배를 저장하였다. 현재까지 상업용 저장고 규모(2톤)로 구축된 자동측정 시스템을 이용하여 저장고내의 온, 습도 및 가스변화를 측정한 data는 <http://203.237.178.103/nong2/index.html>에 누적되어 있다. 자동 측정 시스템의 오차범위를 최소화하고 신뢰성있는 데이터의 확보를 위하여 저장고내의 온·습도를 자동 온·습도 기록계를 이용하여 측정 후 시스템 데이터와 지속적으로 비교하여 함으로써 오차범위를 최소화하였다. 또한 이산화탄소와 산소의 경우는 표준가스(이산화탄소 10.9%, 산소 10.0%)를 이용하여 센서에서 나오는 데이터와 표준가스 농도를 비교하여 자동측정 시스템의 오차범위를 최소화하였다. 센서오차범위 최소화 방안으로 지금까지 수행했던 Gas Chromatography(GC)를 이용한 저장고내의 가스 농도 측정 데이터는 센서값과 비교의 의미가 없다. 이는 GC 측정을 위해 뽑는 기체가 저장고 내부의 공기를 대표한다고 보기 어렵기 때문이다. 따라서 상업용 규모의 저장고내에서는 표준가스를 이용하여 센서에서 나오는 데이터를 보정했다. 또한 온도 습도의 경우 저장고내에 온도 습도계가 마련되어 있어 센서값과의 비교가 가능하므로 일정기간 동안 지속적으로 데이터 비교 관찰한 결과 센서값과 수동 측정값과의 오차가 일정하게 났고 그 오차가 센서의 오차범위 밖에서 일어났으므로 그 차이만큼을 프로그램으로 보정하였다.



# SUMMARY

## I. Title

Automatic Monitoring System for Cool Stores in Postharvest Horticultural Products

## II. Objectives and importance

The yearly amount of postharvest losses are estimated about 30-40% in horticultural products produced in S/Korea. It cannot be easily reduced the losses because of not having enough cold stores around the country. A statistical report of the year 2000 showed the cold storage capacity of 250,000 ton. However the lack of management of cold stores led to an increasing loss of fresh produce. The managing person needs some information of proper storage condition and changing storage factors for each product. Most cold stores have now some difficulties of minimizing the temperature difference under 1-2°C in one room and therefore many managers let the storage temperature maintain the level of 2-3°C more than at the optimum levels. The precise temperature controlling and the minimizing temperature difference can reduce the postharvest losses. In practice humidity as well as temperature should be controlled for maintaining optimum fruit and vegetable quality during storage.

The unexpected storage atmospheres such as the improper temperature, low relative humidity, too high carbon dioxide, and low oxygen can be suddenly formed at the first stage of whole storage period. Who want to store the fresh produce need relevant information of such changing atmospheres. It can be found different physiological disorders such as abnormal ripening, rapid senescence, internal browning, and decay formation in the case of choosing wrong storage methods. Therefore, it is important to develop the automatic monitoring system to avoid different physiological disorders and to reduce postharvest losses. A new automatic

monitoring system development which can manage the environment information accumulates the data for the change of density of oxygen, dioxide, relative humidity, and temperature after sending to the computer in the environment of a low-temperature storage can contribute 1) to the maintenance of optimal storage factors of temperature, relative humidity, oxygen and carbon dioxide, 2) to the prediction of different physiological disorders during ripening and senescence, 3) to the reduction of postharvest losses attacked by microorganism resulting decay, 4) the easy management of storage information every time and everywhere.

It is the aim of this research to conduct the development of the automatic monitoring system to acquire the storage information using different sensors in stores and to record their data in computer automatically and to monitor storage atmospheres of fresh horticultural produce.

### III. Contents and scope

1. Investigation of postharvest physiological characteristics in horticultural products.
  - 1) postharvest physiology in several horticultural crops and application possibility
  - 2) physiological characteristics according to the storage temperatures and produce
  - 3) development of storability evaluation model for each produce
  
2. The development of the automatic monitoring system and information recording of storage atmospheres in fresh horticultural produce.
  - 1) the development of the automatic monitoring system and recording
  - 2) minimizing of error range between setting storage factors and recording data automatically
  - 3) a new improved automatic monitoring system

3. A new automatically integrated monitoring system applicable for commercial stores

#### IV. Results and proposals for practical use

1. Investigation of postharvest physiological characteristics in horticultural products.

Fuji apple fruit showed a typical climacteric pattern with respiration characteristics. The respiration intensity increased until 15 days after storage at 4°C and 10°C, and thereafter decreased gradually until 45 days when no distinct changes were found after that time. Ethylene and ethanol contents were proportionally more at higher storage temperatures in Fuji apples. Niitaka pear fruit were revealed as a non-climacteric fruit after investigating respiration characteristics.

2. The development of the automatic monitoring system and information recording of storage atmospheres in fresh horticultural produce.

- 1) the development of the automatic monitoring system and recording

The automatic monitoring system was elucidated as follows; the temperature, relative humidity, carbon dioxide, oxygen, and ethylene were automatically investigated and accumulated at the proper interval using each recording sensor in many storage chambers. For gas sampling a small motor was needed to collect a small amount of gas volumes through the suction lines connected to each storage chamber. This gas samples were directly moved to each sensor through the suction lines. We used one sensor to analyze each storage factor. At this moment gas detecting sensors can record and accumulate the voltage data as the value of percentage or ppm or °C in host computer, and recording processes can be controlled by application program.

2) a new improved automatic monitoring system

The automatic monitoring system using the suction motor had some difficulties, for example we don't know the gas volume of each gas and the changes of storage atmospheres. Because of these difficulties a new improved monitoring system was applied to let each sensor set and detect the storage information in the ground of storage chambers.

3) A new automatically integrated monitoring system applicable for commercial stores

Our software and hardware technology with experimental devices which were developed for 2 years have been subsequently applied to the new automatically integrated monitoring system applicable for commercial stores in third year. Pilot stores containing the storage capacity of 2-ton apple and pear separately had different sensors of temperature, relative humidity, carbon dioxide, oxygen, and ethylene. There can be found all records and information accumulated automatically in computer site of <http://203.237.178.103/nong2/index.html>. To minimize of error range between setting storage factors and recording data automatically, and to acquire more reliable information, we have been consistently compared both of manual investigation and automatically input data in case of gas concentrations associated with postharvest physiological characteristics. And our efforts to minimize the error range between setting concentrations and recording data automatically have been often done by compensating using the standard gases (CO<sub>2</sub>, O<sub>2</sub>). Gas chromatography (GC) has been used in experimental analysis, but it can't be applied to the new monitoring system due to a great difference of both values. The ppm value and concentration analyzed by GC can't represent the internal concentrations in commercial stores. Improved sensors to assess quality are still needed, and the automatic monitoring system will contribute much to improve the reduction of postharvest losses in horticultural products in coming years.

# CONTENTS

Chapter 1. Introduction .....	1
Section 1. Objective and scope of research .....	1
Section 2. Needs of research .....	2
Section 3. Research trends in domestic and foreign country .....	5
Chapter 2. Postharvest physiological characteristics and automatic monitoring system in horticultural products .....	7
Section 1. Introduction .....	7
Section 2. Postharvest physiological characteristics in horticultural products .....	8
1. Postharvest physiological characteristics .....	8
1) Materials and methods .....	8
2) Results and discussion .....	10
2. Basic modelling for determining a storability .....	26
1) Materials and methods .....	26
2) Results and methods .....	29
3. Establishment of automatic monitoring system and measurements of their information .....	36
1) Experimental system device using a small suction motor .....	36
2) Improved automatic measuring system .....	39
3) Development of applicable automatic measuring system for a commercial store .....	45
Section 3. Conclusions .....	46
Chapter 3. Development of application software program for an automatic monitoring system in storage environment .....	47

Section 1. Introduction .....	47
Section 2. Examination of standard method in measuring systems .....	48
1. Measuring item .....	48
2. Definition of measuring data .....	48
3. Extendability of system in storage and sensor .....	49
4. Definition of measuring interval .....	51
5. Convenience of data analysis .....	51
6. Preference of active storage .....	52
7. Deletion of menubar and toolbar .....	52
8. Convenience for continuous observation .....	52
Section 3. Composition of menu in functions .....	53
1. Local system .....	53
1) Definition of measuring interval .....	54
2) Data view .....	55
3) Graphic view .....	60
4) Preference of storage system .....	62
2. Remote system .....	62
1) Storage system .....	64
2) Data view .....	65
3) Graphic view .....	66
4) Preference of measuring time .....	68
5) Reference room and board .....	70
Section 4. Conclusions .....	70

Chapter 4. Development of Sensor Controller and Monitoring H/W for a Environment Management System .....

Section 1. Introduction .....	73
Section 2. Configuration of sensor modules .....	74
1. Temperature/Humidity sensor .....	75
2. O <sub>2</sub> (Oxygen) sensor .....	76

3. CO <sub>2</sub> (Carbon dioxide) sensor .....	77
4. C <sub>2</sub> H <sub>4</sub> (Ethylene) sensor .....	78
5. Connection of sensor .....	78
1). Operation of 2 wires method .....	78
2). Operation of 4 wires method .....	80
Section 3. Configuration of sensor controller .....	82
1. Input of sensor module .....	82
1). RCV420 .....	82
2). LM741 .....	83
2. ADC Part .....	83
3. Control Part .....	83
4. ROM .....	83
5. Power Part .....	84
6. Data Communication Part .....	84
Section 4. Configuration of middle terminal .....	86
1. Control Part .....	86
2. Data Communication Part .....	86
3. ROM .....	87
4. Analog Multiplexer Part .....	87
5. Power Part .....	87
6. Flowchart .....	88
Section 5. Configuration of communication protocol .....	89
Section 6. Conclusions .....	91
References .....	92
Appendix .....	100

# 목 차

제1장 서론 .....	1
제1절 연구개발의 목적과 범위 .....	1
제2절 연구개발의 필요성 .....	2
제3절 국내·외 연구동향 .....	5
제2장 원예작물의 저장특성 구명 및 환경관리 시스템 개발 .....	7
제1절 서설 .....	7
제2절 원예작물의 수확 후 생리특성 조사 .....	8
1. 생리특성 조사 .....	8
가. 실험방법 .....	8
나. 실험결과 .....	10
2. 작물에 따른 저장성 판단 모델 정립 .....	26
가. 실험방법 .....	26
나. 실험결과 .....	29
3. 원예작물의 환경관리 자동측정시스템의 구축 및 정보측정 .....	36
가. 모터 흡입장치를 이용한 시스템 구축 .....	36
나. 개선된 방식의 자동 측정시스템 개발 .....	39
다. 상업용 저장고 규모에 따른 환경관리의 통일된 시스템을 개발 .....	45
제3절 결론 .....	46
제3장 환경관리를 위한 자동측정 응용 소프트웨어 개발 .....	47
제1절 서설 .....	47
제2절 측정 표준안 작성 .....	48
1. 측정 항목의 종류 .....	48
2. 측정 데이터 정의 .....	48
3. 저장고 및 내부 센서 등 시스템 확장성 고려 .....	49



4. 측정시간 간격 정의 .....	51
5. 데이터 분석의 편리성 고려 .....	51
6. 활성화된 저장고 설정 .....	52
7. 메뉴바와 툴바의 삭제 .....	52
8. 지속적인 관찰을 위한 편리성 고려 .....	52
제3절 기능을 고려한 각 메뉴별 구성 .....	53
1. 로컬 시스템 (Visual C++) .....	53
가. 측정시간 간격 설정 .....	54
나. 데이터 보기 .....	55
다. 그래프 보기 .....	60
라. 저장고 설정 .....	62
2. 원격 시스템 (Web Page) .....	62
가. 저장고에 대하여 .....	64
나. 데이터 보기 .....	65
다. 그래프 보기 .....	66
라. 측정 시간 설정 .....	68
마. 자료방 및 게시판 .....	70
제4절 결론 .....	70
제4장 환경관리 시스템용 센서제어기 및 모니터링 하드웨어 개발 .....	73
제1절 서설 .....	73
제2절 센서부의 구성 .....	74
1. 온/습도 센서 .....	75
2. O <sub>2</sub> (산소) 센서 .....	76
3. CO <sub>2</sub> (이산화탄소) 센서 .....	77
4. C <sub>2</sub> H <sub>4</sub> (에틸렌) 센서 .....	78
5. 센서 연결 .....	78
가. 2 선식 방법의 동작 방법 .....	78
나. 4 선식 방법의 동작 방법 .....	80

제3절 센서 제어기의 구성 .....	82
1. 센서모듈 입력부 .....	82
가. RCV420 .....	82
나. LM741 .....	83
2. ADC부 .....	83
3. 제어부 .....	83
4. ROM .....	83
5. 전원부 .....	84
6. 데이터 송수신부 .....	84
제4절 중계기의 구성 .....	86
1. 제어부 .....	86
2. 데이터 송수신부 .....	86
3. ROM .....	87
4. 아날로그 멀티플렉서부 .....	87
5. 전원부 .....	87
6. 순서도 .....	88
제5절 통신 프로토콜의 구성 .....	89
제6절 결론 .....	91
참고문헌 .....	92
부록 .....	100

# 제 1 장 서 론

## 제1절 연구개발의 목적과 범위

### 1. 연구의 목적

원예산물의 수확후 저장 및 유통과정에서 발생하는 손실은 우리나라 총 생산량의 30-40% 정도로 추산되며 저장고가 충분히 설치되지 않고 수확후 관리 유통관행상 이런 손실율을 줄이기는 쉽지 않다. 현재 농가의 저온저장고의 면적이 증가하는 추세임에도 불구하고 작물의 저온저장에 따른 환경조성과 저장요인의 변화를 수시로 관리, 예측하지 못하는데 따른 피해가 해마다 증가하고 있다. 대형 저장고의 경우 단일 저장고내 수십톤에서 수백톤이 저장되는 현실인데 이 저장고내 온도의 심한 편차 (최소 2-3°C 정도) 때문에 실제로 관리자의 입장에서 적정 온도의 유지란 큰 부담이 되므로 많은 경우 비교적 온도를 높게 유지하는 경우가 관행으로 되어 있다. 따라서 이 경우 정확한 온도센서를 배치한 뒤 컴퓨터에서 자동 관리하게 된다면 저온장해로 인한 손실은 크게 예방하게 되리라 본다.

저온저장고내 환경중 온도, 습도 다음으로 산소의 부족과 과도한 이산화탄소의 집적에 따른 피해가 과수 및 채소에서는 우려되는 상황이다. 이것은 작물에 따른 적정 산소 및 이산화탄소 농도에 대한 정보가 있어야 관리가 가능하지만 실제 저온 저장시 이 가스에 대한 인식은 농가에 거의 알려져 있지 않은 실정이다. 수확 후 발견되는 생리적 장해현상으로 숙성의 빠른 진행이나 노화 그리고 저장고내 이산화탄소의 집적이나 산소의 결핍으로 해석되는 내부갈변 현상 등이 대표적인데 이런 장해발생을 조기 예측하기 위하여 환경관리 자동화 시스템의 개발은 매우 중요하다. 저온저장고내 환경, 즉 온도, 상대습도, 이산화탄소 그리고 산소에 대한 농도변화에 대한 정보를 컴퓨터에 전송한 뒤 자료를 축적해 함으로써 이러한 환경정보를 관리할 수 있는 자동화시스템의 개발은 다음과 같은 여러 가지 부가가치를 제공하여 줄 수 있을 것이다; 1) 최적 온도, 습도, 이산화탄소, 산소 등의 저장 환경인자를 최적 조절 상태로 유지시킬 수 있고, 2) 작물에 따른 특이한 수확후 생리적 숙성 및 노화과정에 대한 변화를 용이하게 예측하게 하고, 3) 부패 미생물에 의한 피해를 최소화하며, 4) 온도 및 습도 유지에 관한 정보를 관리사무실이나 집에서 쉽게 관찰할 수 있는 관리상 편리성을 제공할 수 있을 것이다.

본 연구에서는 원예작물의 저장고 내 품질관정을 목적으로 이와 관련된 여러 환경요인들을 일차적으로 센서에서 감지하고 그 정보를 컴퓨터에 자동 입력하여 이러한 환경을 자동으로 관리할 수 있는 시스템의 개발을 목적으로 하고 있다.

## 2. 연구개발 내용 및 범위

### 가. 원예작물의 수확 후 생리특성 조사

- 1) 원예작물의 품목차이에 따른 수확후 생리 및 적용성 여부 조사
- 2) 작물 및 저장온도에 따른 생리특성 연구
- 3) 작물에 따른 저장성 판단 모델정립

### 나. 원예작물의 환경관리 자동 측정 시스템의 구축 및 정보 측정

- 1) 자동 측정 시스템의 구축 및 측정
- 2) 저장고의 설정기준치와 자동측정 시스템간의 오차범위 최소화
- 3) 개선된 방식의 자동 측정 시스템 개발

### 다. 상업용 저장고의 규모에 다른 환경관리의 통일된 시스템을 개발

## 제2절 연구개발의 필요성

### 1. 기술적 측면

- 저온저장고내 환경을 정확히 관리하는 일은 여러 분야의 기술적 협력 구축으로 가능하다. 우선 수확후 생리학을 기초로 저장작물마다의 최적 저장 환경을 설정해야 하고 저장고내 온도 편차를 최대한 줄일 수 있는 관리적 측면에서 시스템의 보완이 요구된다.
- 채소류중 양파의 경우 1997년 2-3월 심한 경우 부패율이 30-40%까지 발생하였고 이러한 양파와 마늘에서의 부패 증가는 매년 심해지는 추세이다.
- 우선 부패가 심해지는 원인은 예건(큐어링)과 예냉기술의 적립되지 않기 때문으로 해

석되고, 그 외에 중요한 것은 저온 저장고내 환경변화에 대한 정확한 정보를 지속적으로 관찰하지 못하는 데에 기인한 것으로 이는 자동화 관리시스템의 부재에 따른 것으로 이해된다.

- 과수류도 적정 저장온도 이하에서 저온에 의하여 쉽게 저온장해를 받고 있다. 그러나 대형 저장고의 경우 단일 저장고내 수십톤에서 수백톤이 저장되는 현실인데 이 공간내 온도의 심한 편차 (최소 2-3°C정도) 때문에 실제로 관리자의 입장에서 적정 온도의 유지란 어렵다. 이를 피하기 위하여 농가에서는 최적온도보다 비교적 높은 온도를 유지하는 경우가 관행으로 되어 있다. 따라서 이 경우 4-5개를 이용한 정확한 온도센서를 배치한 뒤 컴퓨터에서 자동 관리하게 된다면 저온장해로 인한 손실은 크게 예방될 것이다.
- 일차적으로 국외에서 개발된 센서를 도입하여 연구에 이용하고 이러한 센서를 이차적으로 개발하는 것이 순서이며 이러한 측정장치의 개발은 국내에서 수확후 생리학을 체계적으로 연구하는데 크게 이바지할 수 있을 것이다. 또한 측정기기의 개발은 저장작물의 상태를 산출하여 컴퓨터상에 그래프로 나타나는 시스템의 개발로 이어질 것이다.
- 과수류의 숙성중 생리변화에 대한 구체적 정보는 저온저장 기술뿐만 아니라 CA 저장시 정확한 조건 설정에도 도움이 되는데 이때 산소농도의 조직내 분포 및 이산화탄소의 축적분석이 전제되어야 하는 것이다.
- 저온저장고내 환경을 정확히 관리하는 시스템 개발은 계측기와 연계된 센서 제어기 (Sensor Controller) 및 모니터링 시스템(Monitoring System)의 연결 네트워크를 위한 하드웨어 기술과 계측된 정보의 효율적 분석 및 알고리즘 개발을 위한 응용 소프트웨어(데이터베이스) 기술의 접합으로 실현 가능하다. 우선 원예산물의 수확 후 생리학을 기초로 저장조직의 숙성중 생리적 변화를 수치화한 후 축적하고, 관능 검사를 통한 품질 평가의 결과와 동시에 비교한다.
- 정확한 센서의 측정치가 자동적으로 컴퓨터 상에 기록될 수 있도록 센서는 저장중인

원예작물에서 발생하는 변화를 감지하여 그 데이터를 센서 제어기로 전송하고 여러 센서로부터 전송된 데이터는 저장 상태 모니터링 시스템으로 시리얼(Serial) 네트워크인 RS-232와 RS-422을 통하여 전송된다. 전송된 데이터를 데이터베이스 프로그램에 그래픽 사용자 인터페이스(GUI) 기법을 적용하여 작성한다. 이 때 환경조건에 따라 경고메시지를 내보내어 바로 응급 조치할 수 있도록 구성한다. 이것은 수동적 처리방법이지만 즉각 저장 환경의 변화를 이룸으로서 지속적인 저장의 손실을 극소화가 가능하다.

## 2. 경제·산업적 측면

- 주요 원예작물의 저장중 품질관리로 국제경쟁력 제고를 이룩하여 수출증대 효과를 기대할 수 있다.
- 원예작물의 수확후 상품성 예측을 통한 저장손실의 최소화를 이룩하여 부가가치를 창출할 수 있다.
- 저장성 관리 및 이에 따른 예측시스템의 개발은 많은 원예작물의 수확후 관리에도 비슷하게 응용이 되어 이와 관련된 소프트웨어를 개발한 뒤 국내,외 특허권을 획득할 수 있을 뿐만 아니라 국내 저장산업 및 수출산업의 경쟁력 강화에 이바지 할 것이다.
- 국내에 현재 보급되어 있는 25개의 CA 저장산업체를 포함한 수백개의 저온저장고의 운영을 전산화하는데 이바지할 뿐만 아니라 저장고 구축에 필요한 각종 주변기기의 국산화에 학제간 연구로 발전, 가능할 것이다.

## 3. 사회·문화적 측면

- 과수 및 채소 시장의 개방으로 생산가격의 증가 때문에 국내 농산물의 경쟁력이 약화되고 있을 뿐만 아니라 중국의 WTO 가입이 거론되고 있는 시점에서 국내 농산물의 품질 향상을 위한 종합적인 기술적 지원이 요구되고 있다. 우선 생산에 따른 고품질의 생산에 주력해야 하고 다음 최상의 품질을 유지하기 위한 관리시스템의 개발이 전제되어야 한다.

- 사과, 배, 단감 등의 과수류는 저장한 다음해 4월부터 급격히 품질이 떨어져 소비양상이 갈수록 줄어들고 있고 그 결과 충분한 가격도 보장받고 있지 못하다. 특히 ‘후지’ 사과의 경우 4월 이후의 시장 출하율은 전체 생산량의 12% 수준에 머물고 있다. 따라서 수확시 품질상태로 봄과 여름에도 주요 과수류를 유통시킬 수 있다면 소비패턴의 변화를 유도할 수 있고 이 시점에 증가하는 외국 작물에 대한 소비를 억제시킬 수 있을 것이다.
- 저온저장 방식으로 최대의 효과를 얻을 수 있는 기술을 개발하는 일은 수입에 절대적으로 의존하고 있는 CA 시설에 대한 상당한 대체효과를 가져올 것이다.

### 제3절 국내·외 연구동향

#### 1. 관련기술의 현황과 문제점

- 최근 과일의 저장후 선별을 위하여 고가의 국외 선별기를 대체할 수 있는 국산 선별기 개발에 중점적으로 연구되고 있다. 하지만 이것은 저장후 산물의 품질을 시장성에 따라 분류하는 데 이용되는 것으로 저장중인 원예작물의 품질평가에는 우선 저장고내, 즉 과일밖의 정확한 산소농도의 측정이 필수적이다. 더 나아가 과일의 잔여 저장성을 컴퓨터내서 예측하기 위해서는 아주 정밀하고 아주 가는 센서의 도입 및 개발이 병행되어야 한다.
- 저장중의 품질을 정확히 판별하는 기술개발은 국내에서는 아직 확인되고 있지 않다. 다만 외국에서 독일과 네덜란드 그리고 미국을 중심으로 과일내 가스확산 연구를 기초로 하여 원예작물의 내, 외적 품질을 평가하고 저온저장 및 CA 저장시 최적 조건조건에 대한 정보를 제공하려는 시도가 1990년부터 진행되고 있다.
- 저온저장고에 온도와 습도센서 등이 구비되어 있는 반면 산소와 이산화탄소 센서의 설비가 되어 있지 않다. 또한 일부 정밀센서의 경우 전량 외국으로부터의 수입에 의존하고 있다.
- 저온저장에 따른 환경조성과 저장중 저장요인의 변화를 수시로 관리하지 못하여 수확

후 손실량을 정확히 예측하지 못하는데 따른 피해는 저온저장고의 면적이 증가하는 추세임에도 불구하고 피해가 해마다 증가하고 있는 실정이다. 따라서 수확후 생리 연구에 관련한 각종 주변 측정기기의 개발에 따른 연구수행이 시급히 요구된다.

## 2. 향후 전망

- 이 연구를 통하여 우리나라에서 개발된 각종 온도, 습도, 이산화탄소, 산소 등의 정밀 센서를 농산물의 저장 및 유통산업에 적용하여 보다 정밀한 센서의 개발로 이어질 것이고, 저장 및 유통에 관련된 관리 및 저장성 예측 시스템의 국산화 작업을 가속화 시키는 촉매제 역할을 할 것이다.
- 이 환경관리시스템은 콜드체인시스템(cold chain system)의 구축으로 확산될 것이다.
- 저온저장고의 관리시스템의 장착으로 이용효율이 낮았던 기존시설의 이용 극대화과 시설의 안정적 운영을 가능케 할 것이다.



## 제 2장 원예작물의 저장특성 구명 및 환경관리 시스템 개발

### 제 1절 서설

현재 농가 저온저장고의 면적이 증가하는 추세임에도 불구하고 작물의 저온저장에 따른 환경조성과 저장요인의 변화를 수시로 관리, 예측하지 못하는데 따른 피해가 해마다 증가하고 있다. 따라서 수확후 생리 연구에 관련한 각종 주변 측정기기의 개발에 따른 연구수행이 시급히 요구된다. 한편 저장고내 환경을 자동으로 관리하고 조절할 수 있는 시스템의 구축이 국내에서는 전무한 실정임을 감안할 때, 원예작물의 저장고내 품질관정을 목적으로 이와 관련된 여러 환경 요인들을 일차적으로 센서에서 감지하고 그 정보를 컴퓨터에 자동으로 입력하여 궁극적으로 환경요인을 자동으로 관리할 수 있는 시스템 개발이 시급히 이뤄져야 한다. 저온저장고내 환경, 즉 온도, 상대습도, 이산화탄소, 산소 농도 변화에 대한 정보를 컴퓨터에 전송한 뒤 자료를 축적케 함으로써 이러한 환경정보를 관리할 수 있는 자동화시스템의 개발은 최적 온도, 습도, 이산화탄소, 산소, 에틸렌 등의 저장 환경인자를 최적 조절 상태로 유지시킬 수 있고, 작물에 따른 특이한 수확후 생리적 숙성 및 노화과정에 대한 변화를 용이하게 예측하게 하고, 부패 미생물에 의한 피해를 최소화하며, 온도 및 습도 유지에 관한 정보를 관리사무실이나 집에서 쉽게 관찰할 수 있는 관리상 편리성을 제공한다. 원예작물의 저온저장시 온도, 습도, 기체환경 조성 등의 저장 환경인자와 이에 따른 작물의 품질변화를 정확히 예측하고, 나아가 자동 측정 시스템을 이용한 환경관리 자동화 시스템을 개발하기 위해서는 원예작물의 품목차이에 따른 수확후 생리 및 저장특성을 구명하고 그 적용성 여부를 조사하는 과정이 반드시 선행되어야 한다. 따라서 본 연구에서는 일차적으로 원예작물의 저온저장중 생리특성을 구명하고, 나아가 저장고내 품질관정을 목적으로 이와 관련된 여러 환경요인들을 일차적으로 센서에서 감지하고 그 정보를 컴퓨터에 자동 입력하여 이러한 환경을 자동으로 관리할 수 있는 시스템의 개발을 목적으로 하고 있다.

## 제 2절 원예작물의 수확 후 생리특성 조사

### 1. 생리특성 조사

#### 가. 실험방법

본 연구에서는 원예작물의 저장 특성 구명 및 품질 측정인자를 결정하기 위하여 과수류로는 후지 사과와 신고 배를, 채소류로는 서산 마늘과 창녕 양파를 각각 공시 재료로 이용하여 저온저장시 생리특성 및 품질 변화를 조사하였다. 특히 후지 사과와 신고배에서는 4가지 저장온도를 설정하여 각 저장온도 및 저장기간 경과에 따른 호흡률, 에틸렌 발생, 알코올 생성량, 세포벽 분해효소 활성과 같은 생리적 변화를 조사하였으며, 무게감량, 당도, 색도와 같은 품질조사를 아울러 실시하였다. 또한 마늘과 양파에서는 저온저장시 작물의 호흡 변화양상 구명 및 자동측정 시스템 개발을 위한 기초자료를 마련하고자 이산화탄소 및 산소 농도 변화를 측정하였으며, 이에 따른 에틸렌 발생량도 조사하였다. 또한 저장기간 경과에 따른 품질변화를 조사하기 위하여 맹아율, 부패율, 건전과율, 시장성 항목으로 구성된 관능 검사를 실시하였다.

#### 1) 사과, 배

실험재료는 1998년 11월 충북 충주지역에서 수확한 후지 사과와 경기도 안성지역에서 수확한 신고 배를 사용하였다. 비슷한 크기와 무게, 숙성정도를 지닌 건전과를 선별하여 sodium dichloro-isocyanurate(상품명 커버크로, 일신제약) 수용액(1정/2L)에 30분간 침지하여 소독처리 하였다. 소독이 끝난 후 과일을 실온에서 fan을 이용하여 약 2시간 정도 건조시킨 다음, 사과는 3L 부피의 유리 용기에, 배는 7L 부피의 아크릴 용기에 각각 3개씩의 과일을 배치하고, 이들을 각각 0℃, 4℃, 10℃, 20℃의 저장고에 저장하였다. 각 온도 처리당 15개의 반복구를 두었다. 저장온도에 따른 생리적 특성을 조사하기 위하여 호흡률, 에틸렌 발생, 알코올 생성량을 조사하였고, 저장중 작물의 품질변화를 알아보기 위하여 무게감량, 경도, 당도, 색도, panel test와 같은 품질조사를 실시하였다.

#### 2) 마늘, 양파

실험재료는 1999년 6월 경북 군위에서 수확한 재래종 서산 마늘(한지형)과 경남 창녕에서 수확한 창녕 양파를 공시재료로 이용하였다. 수확 당일 실험실로 옮겨와 비슷한 크기와 무게를 지닌 건전과를 선별하여 2일간 통풍이 잘되는 음지에서 예건시킨 후 실험처

리 하였다. 저장을 위하여 7L 부피의 아크릴 용기에 마늘과 양파 각각 1kg씩을 배치하고 이들을 각각 0℃ 저장고에 저장하였으며, 5개의 반복구를 두었다. 0℃에서의 생리적 특성을 조사하기 위하여 호흡률(CO<sub>2</sub>, O<sub>2</sub>)과 에틸렌 발생량을 조사하였고, 품질변화를 알아보기 위하여 맹아율, 곰팡이 발생율, 건전구(healthy bulb)율 및 시장성 항목으로 구성된 관능검사를 실시하였으며 무게감량을 조사하였다. 본 실험에 사용된 구체적인 조사방법은 다음과 같다.

· 무게감량 : 저장전 과실의 중량을 기준으로 저장 후 감소된 무게를 백분율로 환산하여 표시하였다.

· 당도 : 과실에서 추출한 과즙으로 당도계 (Hand Refractometer, Attago, Japan)를 이용하여 측정하였으며 당도의 표시는 ° Brix로 나타내었다.

· 색도 : 과피의 색도 측정은 Chromameter(CR-200, Minolta, Japan)을 사용하였으며 표준광원 상태에서 L, a, b 값을 측정하였다.

· 호흡률 : 각 처리별로 3개체를 선별하여 7L 부피의 아크릴 용기에 옮겨 24시간 동안 밀폐하였다. 이때 생성된 가스 시료를 용기의 상단으로부터 1ml씩 취하여 gas chromatography(TCD, SHIMADZU model 8APF)를 이용하여 분석하였다.

· 에틸렌 및 에탄올 발생량 : 호흡 측정을 위한 시료 채취 1시간 경과 후 아크릴 용기의 상단으로부터 1ml씩 취하여 gas chromatography(FID, SHIMADZU model 8APF)를 이용하여 분석하였다. 분석조건으로서 Column은 SUS column(SHIMADZU model 201-48705-20)을 사용하였으며, detector 온도는 100℃, column 온도는 120℃로 설정하였으며, 운반기체로 헬륨을 사용하였다.

· Polygalacturonase 활성 분석 : 효소추출액 50 $\mu$ l에 증류수 50 $\mu$ l를 첨가하고 여기에 기질과의 반응을 위하여 0.2% polygalacturonic acid와 30℃에서 20분간 반응시켰다. 이 혼합액에 2-cyanoacetamide를 첨가하여 276nm에서 흡광도를 측정하였다.

·  $\beta$ -galactosidase 활성 분석 : 효소추출액 50 $\mu$ l에 phosphate/citrate buffer를 300 $\mu$ l를 첨가하고 여기에 기질과의 반응을 위하여 50 $\mu$ l  $p$ -nitrophenyl- $\beta$ -D-galactopyranoside와 30 $^{\circ}$ C에서 10분간 반응시킨 후 400nm에서 흡광도를 측정하였다.

· Panel test : 3명의 실험원들이 5일 간격으로 마늘과 양파의 멥아울, 부패율, 건전과율, 시장성을 아래와 같은 방법으로 평가하였다.

· 멥아울, 부패율 : 총 개체중 멥아 및 곰팡이 발생 개체가 차지하는 비율을 백분율로 표시하였다.

· 건전과율 : 총 개체중 멥아 및 곰팡이 발생 개체를 제외한 나머지 개체의 비율을 백분율로 표시하였다.

· 시장성 지수 : 종합적 품질을 5단계 지수 (1=very poor, 2=poor, 3=moderate, 4=good, 5=very good)로 나누어 평가하였다.

## 나. 실험결과

### 1) 사과

저장 온도에 따른 후지 사과의 호흡량을 조사한 결과는 그림 2-1과 같다. 전체저장 기간 중 4 $^{\circ}$ C, 10 $^{\circ}$ C, 및 20 $^{\circ}$ C 처리구의 호흡 발생은 전형적인 호흡급등형 과일(climacteric type)의 양상을 나타내었는데, 4 $^{\circ}$ C와 10 $^{\circ}$ C 처리구에서는 저장 15일까지 호흡이 증가하여 저장 15일째 호흡 최고치(climacteric peak)를 나타내다 이후 감소하여 저장 45일째부터는 호흡량에 큰 변화를 보이지 않았다. 20 $^{\circ}$ C 처리구에서는 저장 직후부터 호흡이 크게 증가하여 저장 30일째 호흡 최고치를 나타내다가 30일을 전후로 하여 감소하는 경향을 나타내었다. 한편 0 $^{\circ}$ C 처리구에서는 저장 30일째 호흡 최고치를 나타냈으나 그 값은 처리중 가장 낮았으며, 전체 저장 기간 중 뚜렷한 호흡량의 변화는 관찰되지 않았다. 저장 온도에 따른 후지 사과의 에틸렌 발생률을 살펴보면 (그림 2-2), 전체 저장기간 중 에틸렌 발생은 호흡발생과 비슷한 경향을 나타내었다. 전체적인 에틸렌 생성량은 저장 온도가 높을수록 크게 나타났다. 0 $^{\circ}$ C, 4 $^{\circ}$ C 및 10 $^{\circ}$ C 처리구에서는 저장 20일까지 에틸렌 함량이 증가하여 최대치를 나타내었으며 저장 20일을 기준으로 하여 감소하는 경향을 나타

내었다. 한편 20℃ 처리구에서는 에틸렌 생성의 최대점이 저장 직후 나타났으며 이후 지속적인 함량 감소가 관찰되었다. 20℃ 처리구에서 에틸렌 생성의 최대점이 나타난 시기는 호흡 최대치가 발생한 시점보다 약 30일 정도 빠른 것을 알 수 있었다.

저장온도에 따른 에탄올 발생량을 조사한 결과 (그림 2-3), 저장온도가 높을수록 에탄올 발생도 높게 나타났다. 0℃ 처리구는 전체 저장 기간 중 0.5 $\mu$ l 이하의 낮은 발생량을 나타냈으며, 저장기간별 비교에서 0℃, 4℃, 및 10℃ 처리구는 저장 30일째 에탄올 발생의 최고치를 보이다가 30일 이후 감소하는 경향을 나타냈으며, 20℃ 처리구는 저장 60일째 최고치를 나타냈다. 0℃와 4℃ 처리구는 전체 저장기간 중 3% 이하의 낮은 무게 감량을 나타냈으며 (그림 2-4), 이들 두 처리간의 뚜렷한 유의차는 관찰되지 않았다. 10℃ 처리구는 저장 120일째 6.5% 정도의 무게손실을 나타냈고 20℃ 처리구는 저장 30일 이후 급속한 무게손실이 발생하여 저장 90일째 45%에 가까운 무게손실을 나타냈다.

과피착색의 지표로 사용된 Hunter 'a' 값의 경우 (그림 2-5), 10℃와 20℃ 처리구에서 비교적 높은 값을 나타내 이 온도범위에서 숙성에 의한 과피 착색이 많이 진행됨을 알 수 있었고, 0℃와 4℃ 처리구는 10℃와 20℃에 비해 상대적으로 낮은 값을 나타내었다. 저장온도별로 세포벽 연화효소인 polygalacturonase(PG)와  $\beta$ -galactosidase의 활성을 분석한 결과 (표 2-1), 저장전과 비교하여 PG의 활성은 저장 30일째 증가하는 경향을 나타냈으며 이러한 경향은 저장온도가 높을수록 뚜렷하게 나타났다. 한편  $\beta$ -galactosidase의 경우, 4℃, 10℃ 및 20℃ 온도 처리구에서는 저장 기간 경과 및 저장 온도 증가에 따라 효소활성도 함께 증가하는 경향을 보였으나, 0℃ 처리구에서는 저장 30일째의 효소활성이 저장초기의 값과 비교하여 큰 유의차를 나타내지 않았다.

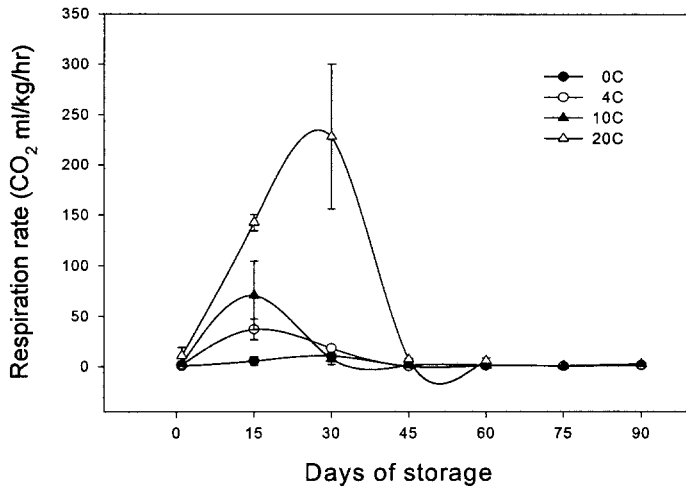


Fig. 2-1. Changes of respiration rate in 'Fuji' apple stored at 0C, 4C, 10C, and 20C for 90 days. Bars show standard deviation.

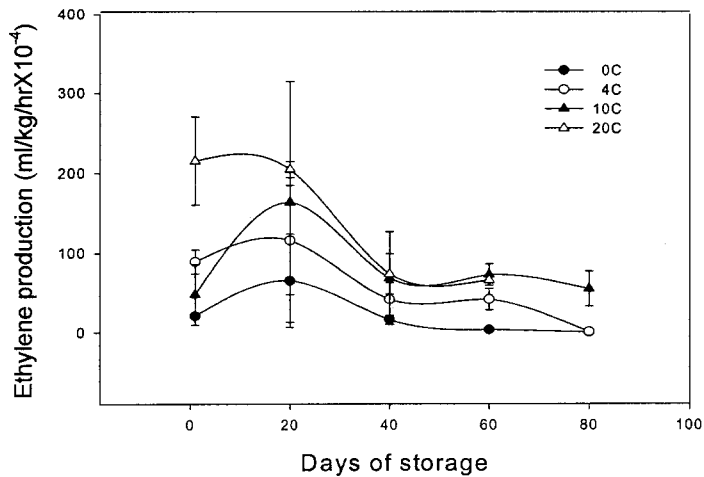


Fig. 2-2. Changes of ethylene production in 'Fuji' apple stored at 0C, 4C, 10C, and 20C for 90 days. Bars show standard deviation.

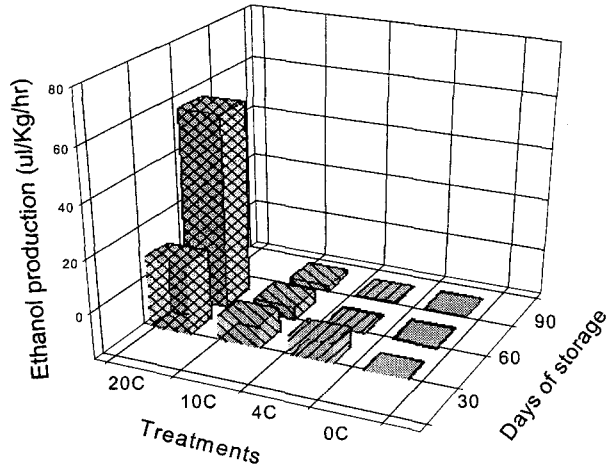


Fig. 2-3. Changes of ethanol production in 'Fuji' apple stored at 0C, 4C, 10C, and 20C for 90 days.

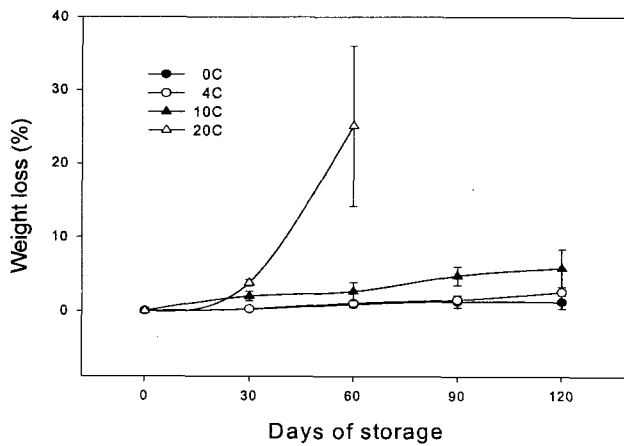


Fig. 2-4. Changes of weight loss in 'Fuji' apple stored at 0C, 4C, 10C and 20C for 90 days. Bars show standard deviation.

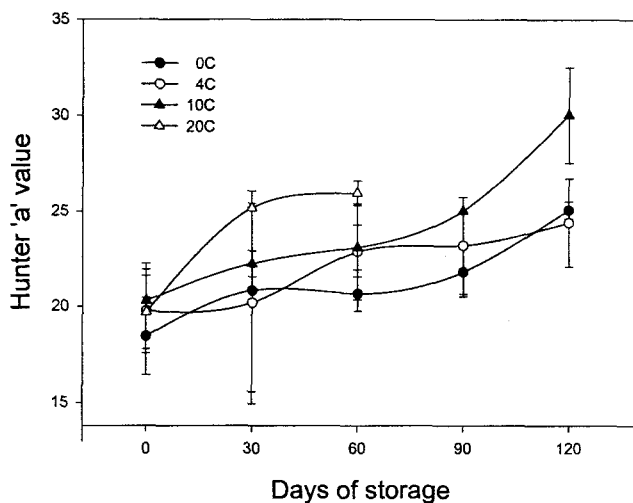


Fig. 2-5. Changes of weight loss in 'Fuji' apple stored at 0C, 4C, 10C, and 20C for 90 days. Bars show standard deviation.

Table 2-1. The Changes of polygalacturonase( $\mu\text{m}$  reducing sugar  $\text{h}^{-1}\text{gfw}^{-1}$ ) and  $\beta$ -galactosidase ( $\text{nmol}$  nitrophenol  $\text{min}^{-1}\text{gfw}^{-1}$ ) activities in 'Fuji' apples during storage at 0°C, 4°C, 10°C, and 20°C.

	Treatments			
	0°C	4°C	10°C	20°C
<b>polygalacturonase</b>				
before storage	710.7			
30 days after storage	1044.8	1135.6	1238.5	1304.9
<b><math>\beta</math>-galactosidase</b>				
before storage	31.2			
30 days after storage	32.4	35.6	37.7	40.6



## 2) 배

저장 온도에 따른 신고 배의 호흡량을 조사한 결과는 그림 2-6과 같다. 전체저장 기간 중 0℃와 4℃ 처리구의 호흡 발생은 비호흡급등형 과일(non-climacteric type)의 양상을 나타내 큰 변화없이 일정하게 낮은 수준을 유지하였다. 10℃ 처리구에서는 저장 30일째 호흡 최대치를 나타내었으며, 30일 이후 호흡이 감소하여 저장 60일째 부터는 일정 수준을 유지하였다. 20℃ 처리구는 저장 직후부터 호흡량이 빠르게 증가하였으며 전체 처리구 중 가장 높은 호흡량을 나타내었다.

신고 배의 에틸렌 발생은 후지 사과에 비해 상대적으로 그 함량이 낮음을 알수 있었다(그림 2-7). 저장 온도에 따른 에틸렌 발생 비교에서 저장 온도가 높았던 10℃와 20℃ 처리구가 저장 온도가 낮았던 0℃와 4℃ 처리구 보다 에틸렌 발생이 높게 나타났으며, 0℃와 4℃ 처리구는 저장 60일째 에틸렌 발생량의 최대치를 보이다 60일 이후 감소하는 경향을 나타내었다.

저장 온도에 따른 신고 배의 에탄올 발생량을 조사한 결과는 그림 2-8과 같다. 에탄올 발생도 에틸렌 발생과 마찬가지로 후지 사과에 비해 그 함량이 적게 나타났다. 20℃를 제외한 온도 처리구의 경우, 전체저장 기간 동안 에탄올 발생은 5 $\mu$ l를 초과하지 않았으며, 저장온도 증가에 따라 에탄올 발생도 증가하는 것으로 나타났으나, 유의차는 크지 않았다.

0℃와 4℃ 처리구는 전체 저장기간 중 2.5% 이하의 낮은 무게 감량을 나타냈으며(그림 2-9), 이들 두 처리간의 뚜렷한 유의차는 관찰되지 않았다. 10℃ 처리구는 저장 120일째 10% 정도의 무게손실을 나타냈고 20℃ 처리구는 저장 직후부터 급속한 무게손실이 발생하여 저장 60일째 15%에 가까운 무게손실을 나타냈다.

Hunter 'L' 값의 경우(그림 2-10), 전체적으로 저장 기간동안 초기값에 비하여 감소하는 경향을 나타냈다. 특히 10℃와 20℃ 처리구는 처리구 중 가장 큰 'L' 값의 감소가 나타나 이 온도범위에서果皮 표면에 흑변이 많이 진행됨을 알 수 있었다.

저장온도별로 세포벽 연화효소인 polygalacturonase(PG)와  $\beta$ -galactosidase의 활성을 분석한 결과(표 2-2), 저장 30일째 PG의 활성은 저장 온도가 높을수록 증가하는 경향을 나타냈으나, 저장기간 경과에 따른 효소 활성 변화에서 20℃ 처리구에서만 유일하게 저장전과 비교하여 효소활성이 증가한 것으로 나타났다. 한편  $\beta$ -galactosidase의 경우, 저장기간 경과 및 저장온도 증가에 따라 효소활성도 증가하는 경향을 나타냈으며 특히 이러한 경향은 20℃ 온도 처리구에서 가장 뚜렷하게 관찰되었다.

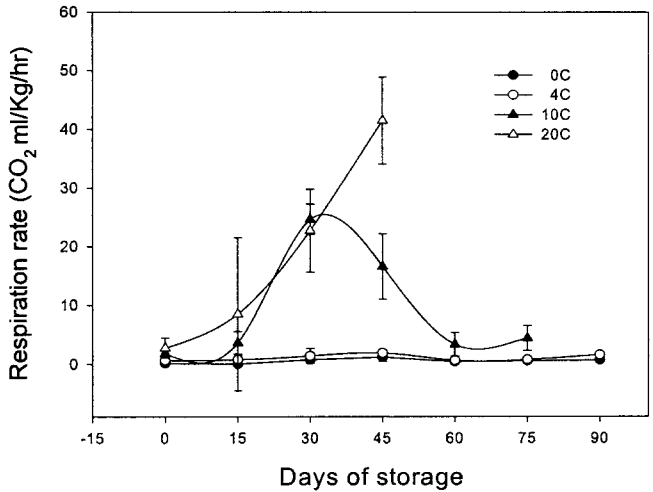


Fig. 2-6. Changes of respiration rate in 'Niiitaka' pear stored at 0C, 4C, 10C and 20C for 90 days. Bars show standard deviation.

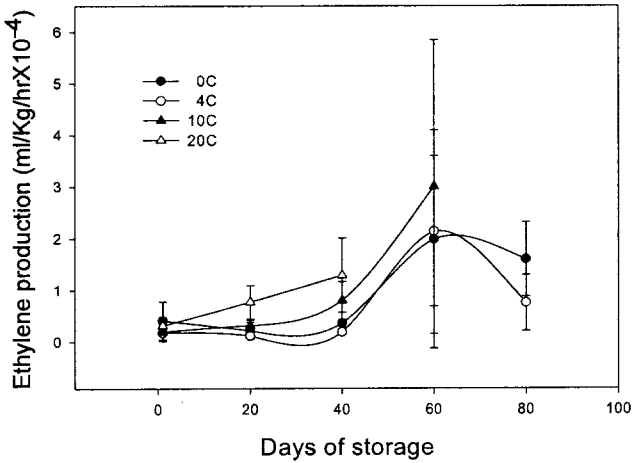


Fig. 2-7. Changes of ethylene production in 'Niiitaka' pear stored at 0C, 4C 10C, and 20C for 90 days. Bars show standard deviation.

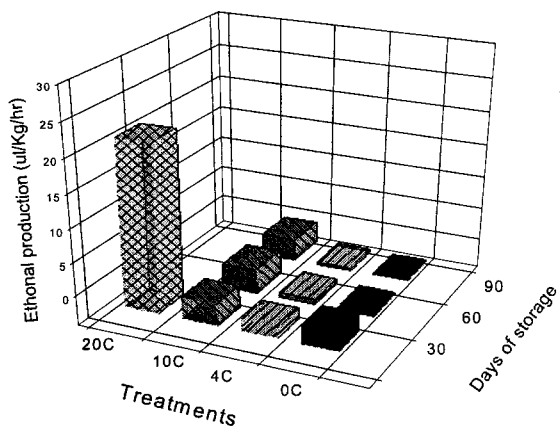


Fig. 2-8. Changes of ethanol production in 'Niiitaka' pear stored at 0C, 4C, 10C, and 20C for 90 days.

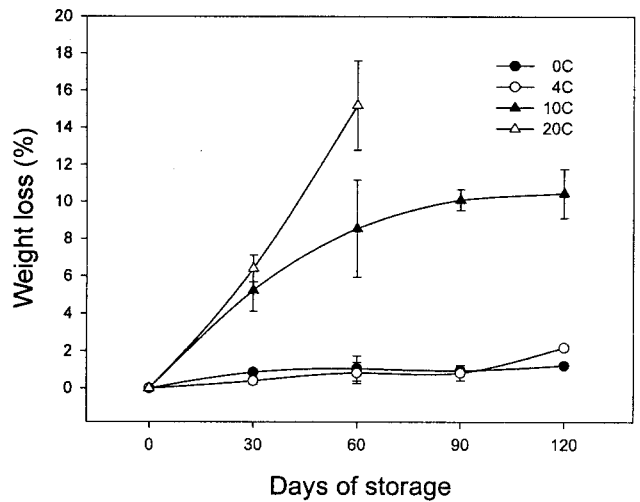


Fig. 2-9. Changes of weight loss in 'Niiitaka' pear stored at 0C, 4C, 10C and 20C for 90 days. Bars show standard deviation.

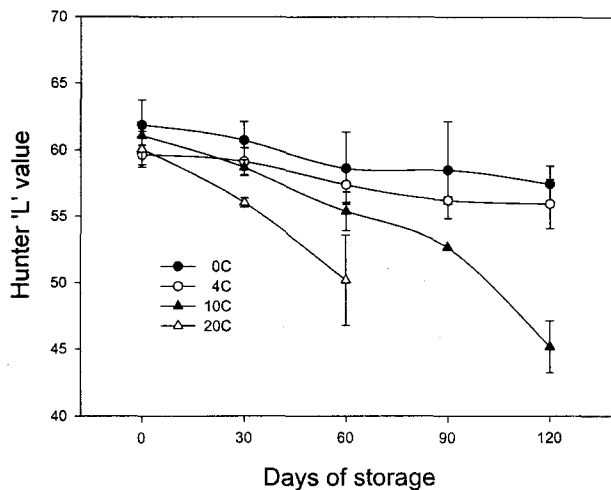


Fig. 2-10. Changes of Hunter 'L' value in 'Niiitaka' pear stored at 0C, 4C, 10C, and 20C for 90 days. Bars show standard deviation.

Table 2-2. The Changes of polygalacturonase( $\mu\text{m}$  reducing sugar  $\text{h}^{-1}\text{gfw}^{-1}$ ) and  $\beta$ -galactosidase ( $\text{nmol}$  nitrophenol  $\text{min}^{-1}\text{gfw}^{-1}$ ) activities in 'Niiitaka' pear during storage at 0°C, 4°C, 10°C, and 20°C.

	Treatments			
	0°C	4°C	10°C	20°C
<b>polygalacturonase</b>				
before storage	1784.6			
30 days after storage	620.3	642.5	737.9	1790.5
<b><math>\beta</math>-galactosidase</b>				
before storage	12.9			
30 days after storage	14.4	13.6	15.0	34.4

### 3) 마늘

그림 2-11은 0℃에서 26일 저장동안 마늘의 무게감량을 나타낸 결과이다. 0℃에서 저장 10일까지 저장전과 비교하여 거의 무게변화를 나타내지 않다가 저장 10일 이후 무게손실이 증가하게 시작하여 저장 26일째 약 4%의 무게 감량을 나타냈다.

0℃에서 저장기간 경과에 따른 마늘의 호흡률( $\text{CO}_2$  ml/Kg/hr)을 살펴본 결과 (그림 2-12), 26일 저장동안 0.9-4.5ml/Kg/hr 수준의 호흡률을 나타냈다. 그러나 저장기간 경과에 따라 호흡 증가 혹은 호흡 감소의 뚜렷한 경향은 나타나지 않았으며, 호흡률이 비교적 낮고, 그 변화의 폭도 크지 않음을 알 수 있었다. 이것은 마늘이 수확 후 호흡량의 변화를 보이지 않거나 혹은 점진적으로 감소하는 호흡 비급등형 (non-climacteric) 작물이기 때문인 것으로 여겨진다. 이 이외에도 0℃ 저장온도에 의해 마늘의 호흡이 억제되었을 것으로 판단되며, 마늘의 장기저장시 생리작용 억제를 위해서는 0℃ 저온저장이 바람직 할 것이라고 판단된다.

0℃에서 저장기간 동안  $\text{O}_2$  농도의 변화로써 호흡률을 나타낸 결과는 그림 2-13와 같다.  $\text{O}_2$  농도는 저장 0일째와 2일째 각각 67.8ml/Kg/hr과 67.0ml/Kg/hr로써 26일 저장동안 최대치를 나타냈으며, 이후 저장 6일까지 지속적으로 감소하다가 6일 이후 66.0-66.3ml/Kg/hr 수준으로 일정하게 유지되었다.

그림 2-14는 0℃에서 저장기간 경과에 따른 마늘의 에틸렌 발생률을 나타낸 결과이다. 저장 0일째와 2일째 각각  $0.15 \mu\text{l/Kg/hr}$ 과  $0.10 \mu\text{l/Kg/hr}$ 로써 26일 저장동안 에틸렌 발생 최대치를 나타냈으며, 이후 저장 4일까지 지속적으로 감소하다가 6일 이후  $0.0045-0.0075 \mu\text{l/Kg/hr}$  수준으로 일정하게 유지되었다. 마늘의 에틸렌 발생량이 극히 적게 나타난 이유는 호흡 비급등형 (non-climacteric) 작물이기 때문인 것으로 판단되며 따라서 저장기간동안 에틸렌의 자촉매적 (auto-catalytic) 생성은 이루어지지 않을 것으로 예상된다.

저장 30일동안 마늘의 멍아울, 곰팡이 발생을, 건전과율 및 시장성을 평가한 결과 (표 2-3), 저장 30일째까지 멍아는 발생하지 않은 반면, 전체 개체의 8.3%에서 곰팡이가 발생하였다. 또한 건전구의 비율도 저장 20일째 100%에서 저장 30일째 91.7%로 감소하였다. 전체적인 시장성을 조사한 항목에서는 저장 20일째까지 시장성이 '매우 우수'로 평가되었으며, 저장 30일째에는 '우수'로 평가되었다.

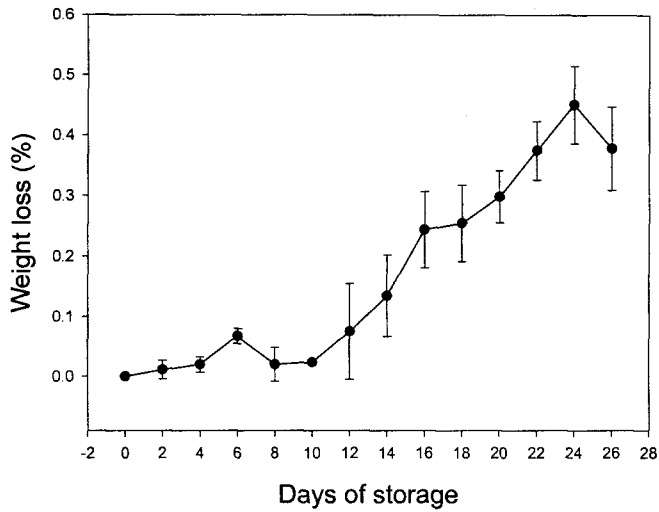


Fig. 2-11. Changes of weight loss in garlic at 0C for 26 days. Bars show standard deviation.

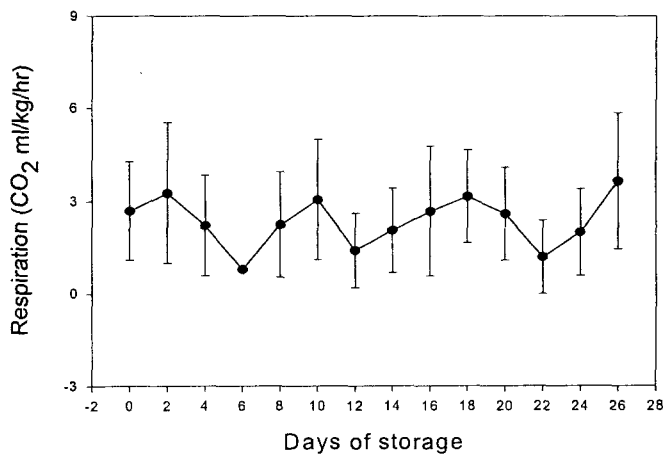


Fig. 2-12. Changes of respiration rate in garlic at 0C for 26 days. Bars show standard deviation.

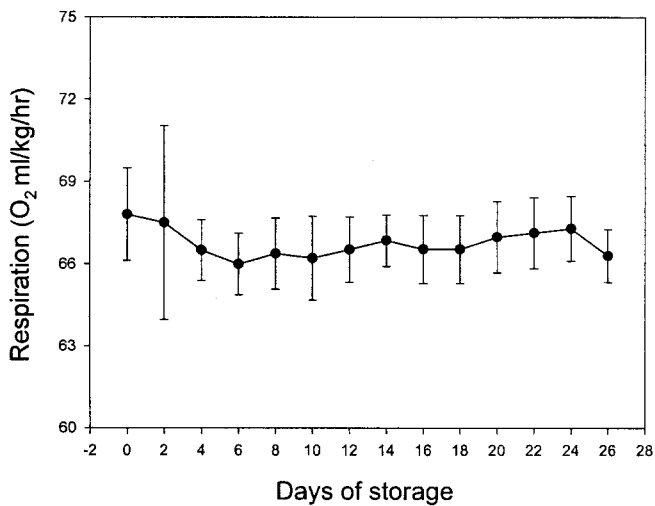


Fig. 2-13. Changes of respiration rate in garlic at 0C for 26 days .  
Bars show standard deviation.

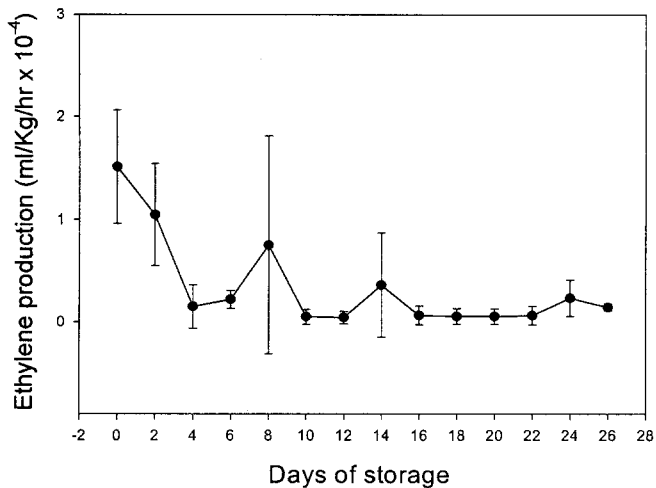


Fig. 2-14. Changes of ethylene production in garlic at 0C for 26 days  
Bars show standard deviation.

Table 2-3. The Changes of sensory evaluation in garlic during storage at 0°C for 30 days.

Days of storage	Sprout(%)	Mold(%)	Healthy bulb(%) <sup>z</sup>	Market quality <sup>y</sup>
before storage	0	0	100	5
5	0	0	100	5
10	0	0	100	5
15	0	0	100	5
20	0	0	100	5
25	0	0	100	4.5
30	0	8.3	91.7	4

<sup>z</sup> 100 - (sprout(%) + mold(%))

<sup>y</sup> 5=excellent, 4=good, 3=moderate, 2=poor, 1=very poor

#### 4) 양 파

그림 2-15는 0°C에서 26일 저장동안 양파의 무게감량을 나타낸 결과이다. 0°C에서 저장 4일까지 저장전과 비교하여 거의 무게변화를 나타내지 않다가 저장 4일 이후 지속적인 무게감량을 나타내, 저장 26일째 약 14%에 이르는 무게 손실을 나타냈다.

0°C에서 저장기간 경과에 따른 양파의 호흡률(CO<sub>2</sub> ml/Kg/hr)을 살펴본 결과 (그림 2-16), 저장 0일과 2일째 각각 0.85ml/Kg/hr와 1.01ml/Kg/hr의 호흡률을 나타내 저장기간중 호흡 최대치를 보였으며, 저장 2일을 기점으로 하여 호흡률은 전반적으로 감소하는 경향을 나타냈다. 또한 양파의 호흡률은 저장 26일동안 0.17-1.01ml/Kg/hr 정도로 낮게 유지되었다. 이것은 양파가 호흡 비급등형 (non-climacteric) 작물이기 때문인 것으로 여겨지나, 수확 후 뚜렷한 호흡량의 변화를 보이지 않은 마늘과는 달리 양파는 저장기간중 점진적으로 호흡이 감소하였으며, 호흡량도 상대적으로 마늘에 비해 적게 나타나 같은 호흡 비급등형 (non-climacteric) 작물에 분류된다 할지라도 작물마다 호흡발생 양상에 다소의 차이가 있는 것으로 여겨진다. 이 이외에도 0°C 저장온도에 의해 양파의 호흡이 억제되었을 것으로 판단된다.

O<sub>2</sub> 농도의 변화로써 양파의 호흡률을 나타낸 결과 (그림 2-17), O<sub>2</sub> 농도는 저장 0일째 41.3ml/Kg/hr로써 26일 저장동안 최대치를 나타냈으며, 이후 저장 6일까지 지속적으로



감소하다가 6일 이후 40.8-40.9ml/Kg/hr 수준으로 일정하게 유지되었다.

그림 2-18은 0℃에서 저장기간 경과에 따른 양파의 에틸렌 발생률을 나타낸 결과이다. 저장 24일까지 에틸렌은 검출되지 않았으며 저장 26일째 2.35 $\mu$ l/kg/hr 정도의 에틸렌 발생량을 나타냈다. 양파의 에틸렌 발생량이 극히 적게 나타난 이유는 양파가 호흡 비급 등형 (non-climacteric) 작물이기 때문인 것으로 판단되며, 따라서 저장기간동안 에틸렌의 자촉매적(autoctalytic) 생성은 이루어지지 않을 것으로 예상된다.

저장 30일동안 양파의 멍아울, 곰팡이 발생율, 건전과율 및 시장성을 평가한 결과 (표 2-4), 저장 30일째까지 조사한 개체중에서 멍아 및 곰팡이는 발생하지 않았다. 또한 건전구의 비율도 저장 30일째 100%를 나타냈다. 전체적인 시장성을 조사한 항목에서는 저장 5일째까지 시장성이 '매우 우수'로 평가되었으며, 저장 10일후부터 저장 30일까지는 '우수'로 평가되었다.

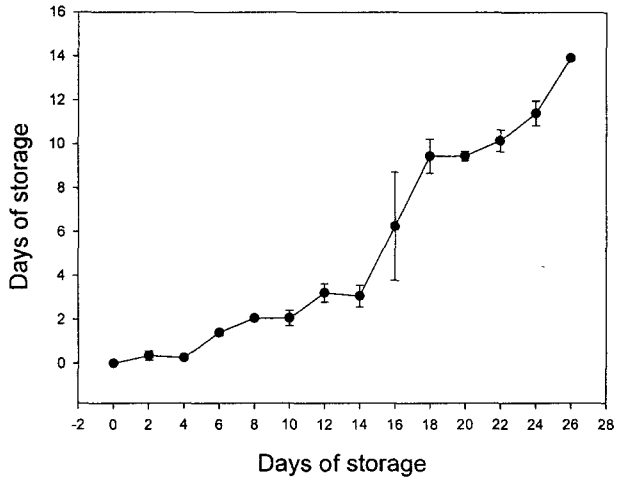


Fig. 2-15. Changes of weight loss in onion at 0°C for 26 days. Bars show standard deviation.

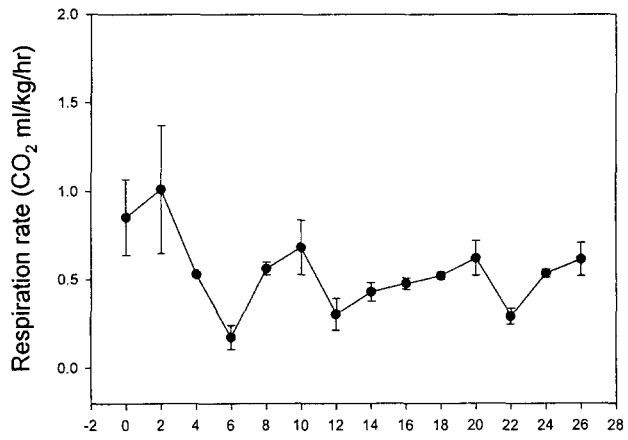


Fig. 2-16. Changes of respiration rate in onion at 0°C for 26 days. Bars show standard deviation.

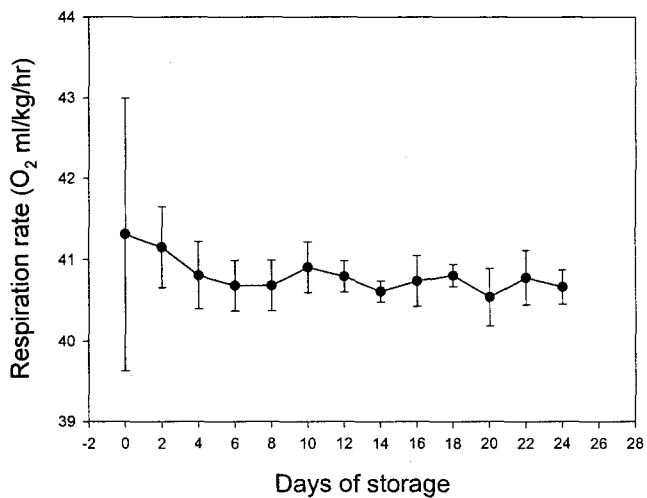


Fig. 2-17. Changes of oxygen concentration in onion at 0C for 26 days  
Bars show standard deviation.

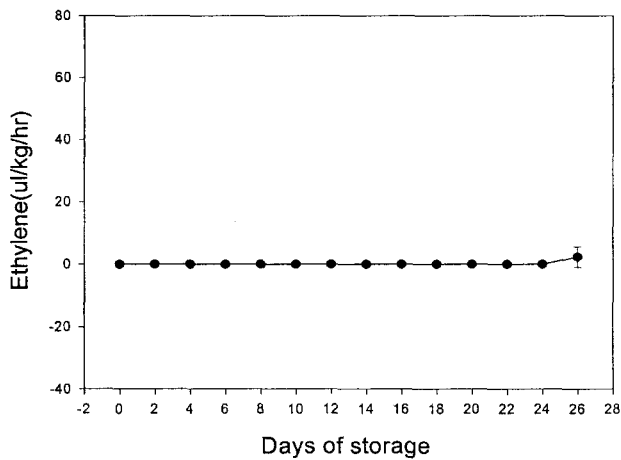


Fig. 2-18. Changes of ethylene production in onion at 0C for 26 days.  
Bars show standard deviation.

Table 2-4. The Changes of sensory evaluation in onion during storage at 0°C for 30 days.

Days of storage	Sprout(%)	Mold(%)	Healthy bulb(%) <sup>z</sup>	Market quality <sup>y</sup>
before storage	0	0	100	5
5	0	0	100	5
10	0	0	100	4.5
15	0	0	100	4.5
20	0	0	100	4.5
25	0	0	100	4.5
30	0	0	100	4.5

<sup>z</sup> 100 - (sprout(%) + mold(%))

<sup>y</sup> 5=excellent, 4=good, 3=moderate, 2=poor, 1=very poor

## 2. 작물에 따른 저장성 판단 모델정립 (상업용 저장고 규모)

### 가. 실험방법

원예산물의 저장성 판단 모델을 정립하기 위하여 상업용 규모의 저온저장에 따른 배와 사과, 마늘과 양파의 생리특성과 품질변화를 데이터화하였다.

#### 1) 사과, 배

실험재료는 2000년 11월 충남 예산에서 수확된 ‘후지’ 사과와 2000년 10월 성환에서 수확된 ‘신고’ 배를 공시재료로 이용하였다. 사과와 배의 저온저장을 위하여 비슷한 크기와 무게, 숙성정도를 지닌 건전과를 선별한 후 2ton 규모 상업용 pilot 저장고 2개에 각각 사과와 배를 저장하였다. 저장온도는 0± 0.5°C로 설정하였고, 저장기간별 생리특성과 품질을 7일 간격으로 조사하였으며, 구체적인 조사방법은 다음과 같다.

· 무게감량 : 저장전 과실의 중량을 기준으로 저장 후 감소된 무게를 백분율로 환산하여 표시하였다.

· 경도: 과실의 과피를 제거한 후 11mm plunger를 장착한 Fruit Hardness

tester(N.O.W., Cat. No. 9300, Japan)를 이용하여 과실의 상단 부위에서 일정한 압력을 가하여 측정 한 뒤 평균값을 kg으로 표시하였다.

· 당도 : 과실에서 추출한 과즙으로 당도계 (Hand Refractometer, Attago, Japan)를 이용하여 측정하였으며 당도의 표시는 Brix° 로 나타내었다.

· 색도 : 과피의 색도 측정은 Chromameter(CR-200, Minolta, Japan)을 사용하였으며 표준광원 상태에서 L, a, b 값을 측정하였다.

· 호흡률 : 각 처리별로 3개체를 선별하여 7L 부피의 아크릴 용기에 옮겨 24시간 동안 밀폐하였다. 이때 생성된 가스 시료를 용기의 상단으로부터 1ml씩 취하여 gas chromatography(TCD, SHIMADZU model 8APF)를 이용하여 분석하였다.

· 에틸렌 발생률 : 호흡 측정을 위한 시료 채취 1시간 경과 후 아크릴 용기의 상단으로부터 1ml씩 취하여 gas chromatography(FID, SHIMADZU model 8APF)를 이용하여 분석하였다. 분석조건으로서 Column은 SUS column(SHIMADZU model 201-48705-20)을 사용하였으며, detector 온도는 100℃, column 온도는 120℃로 설정하였으며, 운반기체로 헬륨을 사용하였다.

· 내부 호흡 및 에틸렌 : 각 처리별로 3개체를 선별하여 내부 호흡 및 에틸렌을 측정하였으며, 이때 사용된 GC 분석조건은 외부 호흡 및 에틸렌 측정과 동일한 방법을 사용하였다.

· Panel test : 3명의 실험원들이 처리별로 맛, 조직감, 부패율, 광택, 모양 등의 평가항목에 관하여 5단계 지수를 사용하여 평가하였다.

## 2) 마늘, 양파

실험재료는 2000년 5월 경북 군위에서 수확한 재래종 서산 마늘(한지형)과 경남 창녕에서 수확한 창녕 양파를 공시재료로 이용하였다. 수확 당일 실험실로 옮겨와 비슷한 크기와 무게를 지닌 건전과를 선별하여 2일간 통풍이 잘되는 음지에서 예건시킨 후 2ton 규

모 상업용 pilot 저장고 2개에 각각 마늘과 양파를 저장하였다. 저장온도는  $0 \pm 0.5^{\circ}\text{C}$ 로 설정하였고, 저장기간별 생리특성과 품질을 7일 간격으로 조사하였으며, 구체적인 조사 방법은 다음과 같다. 생리적 특성을 조사하기 위하여 호흡률( $\text{CO}_2$ ,  $\text{O}_2$ )과 에틸렌 발생량을 조사하였고, 품질변화를 알아보기 위하여 멥아울, 곰팡이 발생, 건전구(healthy bulb) 및 시장성 항목으로 구성된 관능검사를 실시하였으며 무게감량을 조사하였다.

· 무게감량 : 저장전 과실의 중량을 기준으로 저장 후 감소된 무게를 백분율로 환산하여 표시하였다.

· 당도 : 과실에서 추출한 과즙으로 당도계 (Hand Refractometer, Attago, Japan)를 이용하여 측정하였으며 당도의 표시는  $^{\circ}$  Brix로 나타내었다.

· 색도 : 과피의 색도 측정은 Chromameter(CR-200, Minolta, Japan)을 사용하였으며 표준광원 상태에서 L, a, b 값을 측정하였다.

· 호흡률 : 각 처리별로 3개체를 선별하여 7L 부피의 아크릴 용기에 옮겨 24시간 동안 밀폐하였다. 이때 생성된 가스 시료를 용기의 상단으로부터 1ml씩 취하여 gas chromatography(TCD, SHIMADZU model 8APF)를 이용하여 분석하였다.

· 에틸렌 및 에탄올 발생량 : 호흡 측정을 위한 시료 채취 1시간 경과 후 아크릴 용기의 상단으로부터 1ml씩 취하여 gas chromatography(FID, SHIMADZU model 8APF)를 이용하여 분석하였다. 분석조건으로서 Column은 SUS column(SHIMADZU model 201-48705-20)을 사용하였으며, detector 온도는  $100^{\circ}\text{C}$ , column 온도는  $120^{\circ}\text{C}$ 로 설정하였으며, 운반기체로 헬륨을 사용하였다.

· Panel test : 3명의 실험원들이 5일 간격으로 마늘과 양파의 멥아울, 부패율, 건전과율, 시장성을 아래와 같은 방법으로 평가하였다.

· 멥아울, 부패율 : 총 개체중 멥아 및 곰팡이 발생 개체가 차지하는 비율을 백분율로 표시하였다.

· 건전과율 : 총 개체중 멧아 및 곰팡이 발생 개체를 제외한 나머지 개체의 비율을 백분율로 표시하였다.

· 시장성 지수 : 종합적 품질을 5단계 지수 (1=very poor, 2=poor, 3=moderate, 4=good, 5=very good)로 나누어 평가하였다.

## 나. 실험결과

### 1) 사과

저온저장에 따른 후지 사과의 품질변화를 조사한 결과는 그림 2-19와 같다. 저장기간 중 저온 처리구의 호흡 발생 경향은 호흡급등형 과일(climacteric type)의 양상을 나타내었는데, 저장 14주째 4.2ml/Kg/hr로써 호흡 최고치(climacteric peak)를 나타내었다.

에틸렌 발생률은 호흡 발생량과 비슷한 경향을 보였으며, 저장 3주째 에틸렌 발생 최대치가 나타났다.

무게감량은 저장기간에 따라 증가하는 경향을 나타내, 저장 20주째 약 6% 정도의 무게감량을 나타내었다.

과피착색의 지표로 사용된 Hunter 'a' 값의 경우 저장기간 경과에 따라 증가하는 경향을 나타냈다.

경도 및 당도는 각각 저장기간에 따라 감소하거나 (경도), 증가하는 경향(당도)을 나타냈다.

저장온도에 따른 품질요인들을 종합하여 시장성으로 나타낸 결과, 저장 12주째까지 높은 시장성을 유지하였으며, 저장 10주째까지 시장성 지수 4 이상을 나타내 품질이 우수한 것으로 판단되었다.

### 2) 배

저온저장에 따른 신고 배의 품질변화를 조사한 결과는 그림 2-20과 같다. 전체저장 기간 중 두 처리구의 호흡 발생은 비호흡급등형 과일(climacteric type)의 양상을 나타내었는데, 저온처리구의 호흡률은 저장 14일째까지 호흡 발생의 큰 증가없이 일정하게 낮은 수준을 유지하다가 14주 이후 다소 증가하는 경향을 보였다. 신고 배의 에틸렌 발생은 후지 사과에 비해 상대적으로 그 함량이 낮게 나타났으며, 저장후기에 비해 저장초기에 에틸렌 발생이 상대적으로 높게 나타났다.

Hunter 'L' 값의 경우, 저장 기간동안 초기값에 비하여 감소하는 경향을 나타냈으며, 당도는 저장기간 경과에 따라 증가하는 경향을 나타냈다. L값의 감소는 저장기간경과에 따른 과피 흑변현상을 반영한 것으로 판단된다.

품질요인들을 종합하여 시장성으로 나타낸 결과, 저장 10주째까지 높은 시장성을 유지하였으며, 저장 12주째까지 시장성 지수 3 이상을 나타내 품질이 우수한 것으로 판단되었다.

### 3) 마늘

그림 2-21은 저장115일동안 마늘의 곰팡이 발생율을 평가한 결과이다. 저장 45일째까지 곰팡이가 발생하지 않았고, 저장 60일째 전체 개체의 8.9%에서 곰팡이가 발생하였다. 또한 맹아가 발생하지 않은 건전구의 비율도 저장 0일째 100%에서 저장 115일째 45%로 감소하였다. 전체적인 시장성을 조사한 항목에서는 저장 45일째까지 시장성이 '매우 우수'로 평가되었으며, 저장 75일째에는 '보통'으로 평가되었다(자료 미제시).

0℃에서 저장기간 경과에 따른 마늘의 호흡률(CO<sub>2</sub> ml/Kg/hr)을 살펴본 결과 (그림 2-22), 약 12주간의 저장동안 0.9-5.8ml/Kg/hr 수준의 호흡률을 나타냈다. 저장 2주째 부터는 호흡률의 큰 변화를 나타내지 않았다. 이것은 마늘이 수확 후 호흡량의 변화를 보이지 않는 비급등형 (non-climacteric) 작물이기 때문인 것으로 여겨지며, 이 이외에도 0℃ 저장온도에 의해 마늘의 호흡이 억제되었을 것으로 판단된다. 따라서 마늘의 장기저장시 생리작용 억제를 위해서는 0℃ 저온저장이 바람직 할 것이라고 판단된다.

### 4) 양파

저장 15주 동안 양파의 곰팡이 발생율 및 건전과율을 평가한 결과 (그림 2-25), 저장 30일째까지 조사한 개체중에서 맹아 및 곰팡이는 발생하지 않았다. 또한 건전구의 비율도 저장 30일째 100%를 나타냈으나 저장기간 경과에 따라 부패 및 맹아 발생이 증가하여 저장 15주째에는 전체개체의 약 37%정도의 부패율을 나타냈다.

전체적인 시장성을 조사한 항목에서는 저장 30일째까지 시장성이 '우수'로 평가되었으며, 저장 30일부터 시장성이 감소해 저장 95일째 부터는 시장성을 상실한 것으로 평가되었다.

0℃에서 저장기간 경과에 따른 양파의 호흡률(CO<sub>2</sub> ml/Kg/hr)을 살펴본 결과 (그림 2-26), 저장 1주 이후부터 지속적으로 호흡이 증가해 저장말기에는 약 2.24ml/Kg/hr의 호흡률을 나타냈다. 이것은 양파가 호흡 비급등형 (non-climacteric) 작물이기는 하나 부



패 및 맹아 발생에 의해 호흡이 증가한 것으로 판단된다.

그림 2-27는 0℃에서 15주 저장동안 양파의 경도를 나타낸 결과이다. 0℃에서 저장 30 일째부터 지속적인 경도의 감소가 나타나, 저장15주째에는 약 50%에 이르는 경도손실을 나타냈다. 양파의 저장중 L값은 지속적으로 감소되었다(그림 2-28).

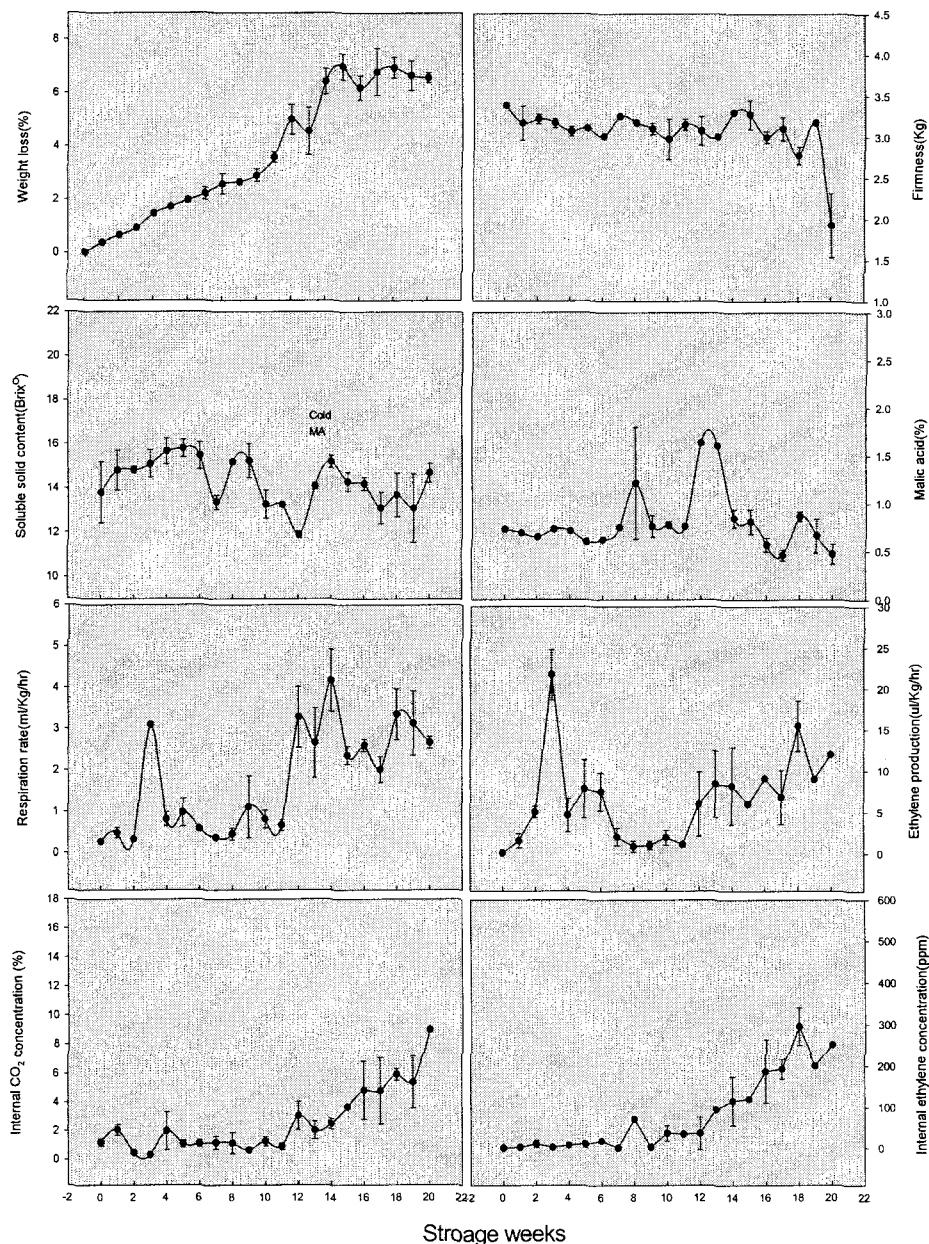


Fig. 2-19. Changes of quality characteristics according to storage days (weeks) in Fuji apple

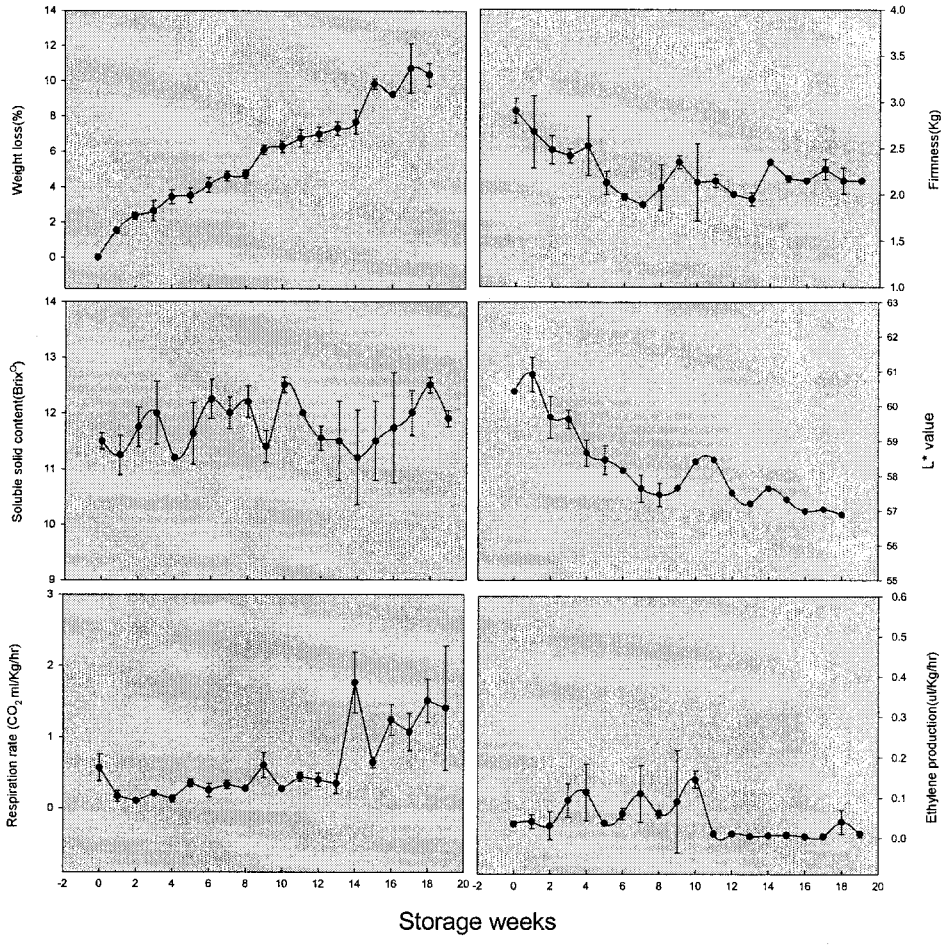


Fig. 2-20. Changes of quality characteristics according to storage days (weeks) in Niitaka pear

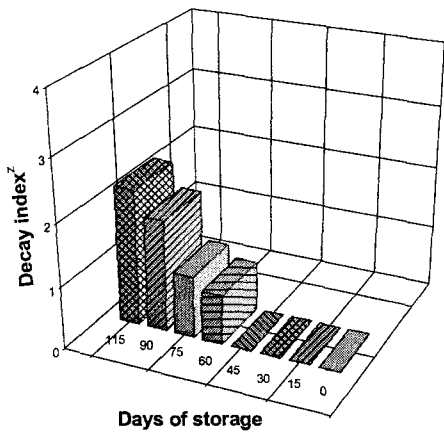


Fig. 2-21. Decay incidence of garlic by cold storage during 115 days of storage.

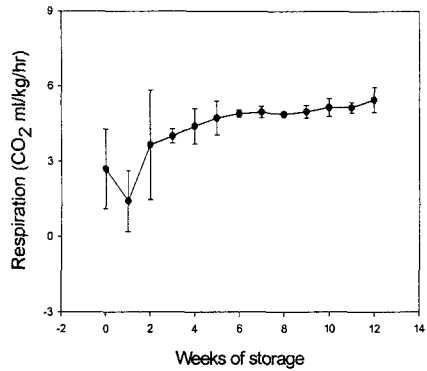


Fig. 2-22. Changes of respiration rate in garlic at 0C for 13 weeks. Bars show standard deviation.

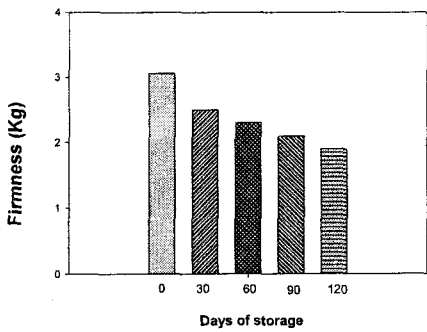


Fig. 2-23. Changes of firmness of garlic by cold storage during 120 days of storage.

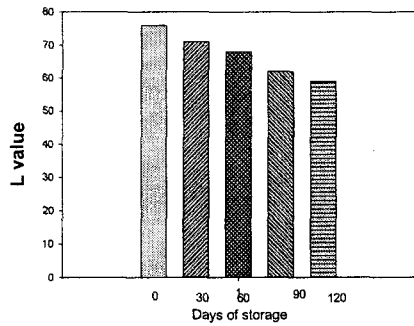


Fig. 2-24. Changes of L value of garlic by cold storage during 120 days.

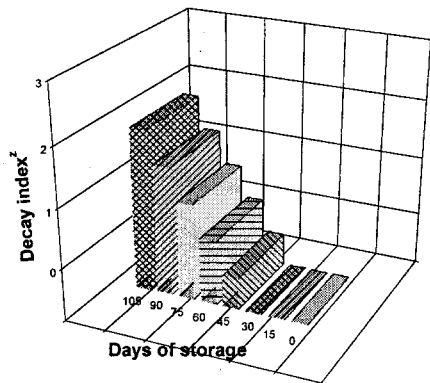


Fig. 2-25. Decay incidence of onion by cold storage during 105 days.

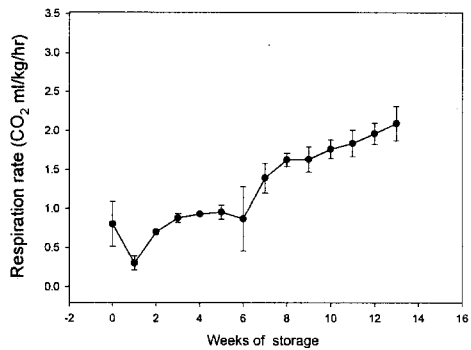


Fig. 2-26. Changes of respiration rate in onion at 0C for 14 weeks. Bars show standard deviation.

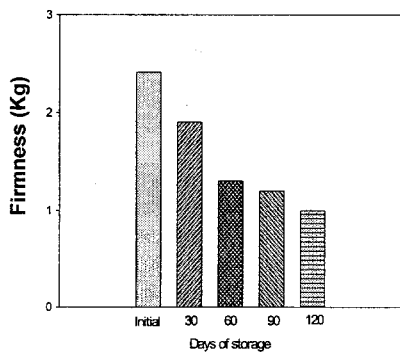


Fig. 2-27. Changes of firmness of onion during cold storage at 120 days.

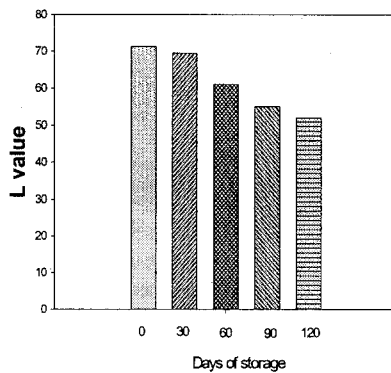
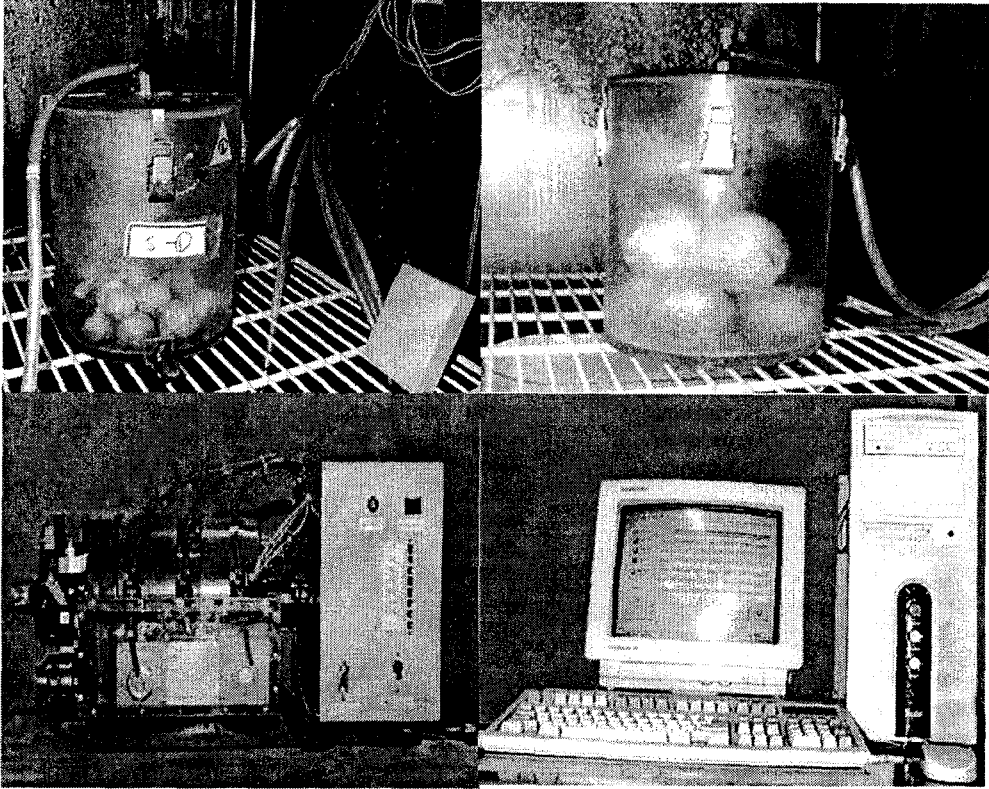


Fig. 2-28. Changes of L value of onion during cold storage at 120 days.

3. 원예작물의 환경관리 자동 측정 시스템의 구축 및 정보 측정  
가. 모터 흡입장치를 이용한 시스템 구축



< 그림 2-19. 자동 측정 시스템의 전체적인 기본 구성 장치 >

1) 시스템 개요

작물이 들어있는 여러개의 저장용기 내부의 온, 습도 및 이산화탄소, 산소, 에틸렌 농도 측정을 위해서 해당하는 각각의 센서를 이용, 환경정보를 용기별로 순차적으로 자동 측정하는 방식이다. 이때 저장용기 내의 가스샘플을 수집하기 위해서 모터를 설치하고, 모터에 의해 저장 용기내의 가스를 일정량 흡입하게 된다. 이때 얻어진 가스샘플은 저장 용기와 연결된 흡입관을 통해 각 센서까지 전달된다. 온, 습도 센서를 비롯한 가스 측정 센서는 종류별로 각 1개씩이 사용되었다. 이때 각 센서를 통해 나온 전류 (혹은 전압) data는 호스트 컴퓨터상의 응용 소프트웨어상에서 사용자가 원하는 단위인 % (습도, 이산화탄소, 산소)나 ppm(에틸렌), 혹은 °C(온도)로 변환되며, 이 모든 과정은 컴퓨터 프로그램상에서 제어가 가능하다

## 2) 자동측정 시스템을 이용한 저장고내 환경정보 측정

시스템에 이용된 저장용기의 규격은 7L 부피의 아크릴 저장 용기를 사용하였으며, 총 6개의 저장 용기중 1번부터 3번 저장 용기에는 마늘을, 4번부터 6번 저장용기에는 양파를 각각 1Kg씩 배치하였다. 저장온도는 0℃로 설정하였으며 데이터의 측정간격은 1시간으로 설정하였다.

## 3) 저장고의 설정기준치와 자동측정 시스템간의 오차범위 최소화

### 가) 온도, 습도

자동측정 시스템을 이용한 신뢰성 있는 데이터를 얻고 측정치간 오차 범위를 최소화하기 위하여 자동 온·습도 기록계를 사용하여 저장고 내의 온·습도를 24시간 간격으로 조사하였으며, 이를 동일한 시간대에 측정된 시스템 데이터와 지속적으로 비교하였다(그림 2-30). 초기 저장고 설정 온도는 0℃였으나, 시스템 및 자동 온·습도 기록계를 통하여 나온 데이터 모두 처음 저장고 설정온도와는 차이가 있었으며, 시스템과 자동 온·습도 기록계 두 측정치간에도 같은 비교시점에서 최소 약 0.02℃에서 최대 1.5℃ 정도의 편차를 나타냈다.

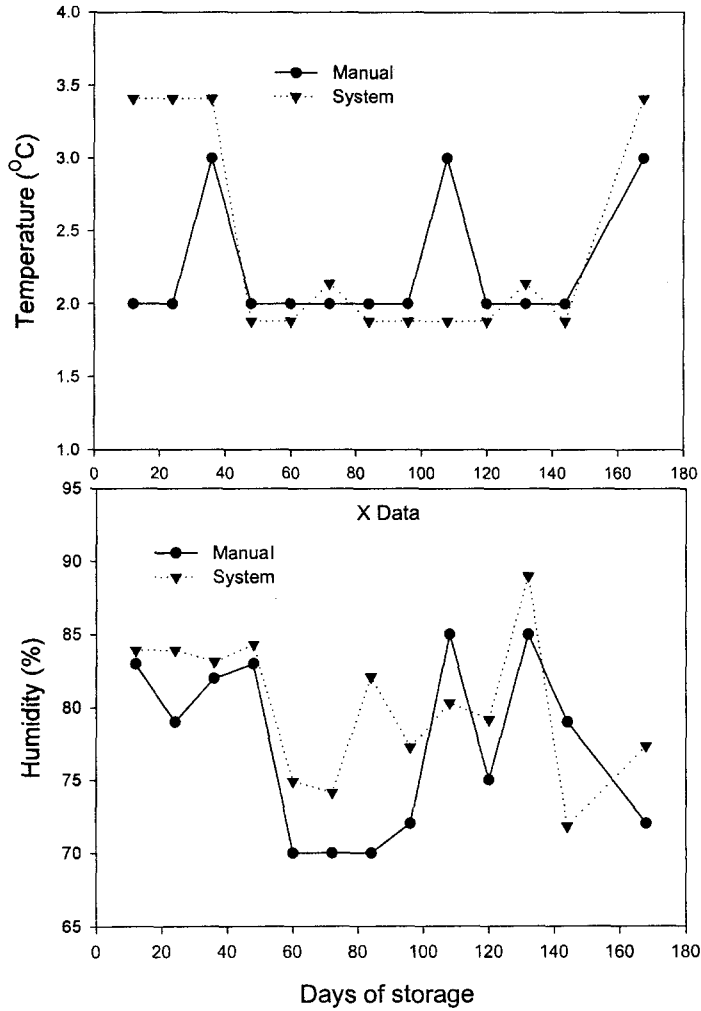


Fig. 2-30. Temperature and humidity data in storage room measured by sensor(system) and electronic termohygrograph(manual).



## (2) 가스 농도

자동측정 시스템을 이용한 신뢰성 있는 데이터를 얻기 위하여 시스템과 연결된 저장용기내의 기체조성을 GC (TCD, FID)를 이용해 12시간 간격으로 조사하였으며, 이를 동일한 시간대에 측정된 시스템 데이터와 지속적으로 비교하였다. 또한 모터에 의해 흡입되는 공기의 양 및 이에 따른 저장 용기내의 가스환경 변화를 알아보기 위하여 저장 용기의 아래 밸브만 열어 놓은 상태에서 용기내의 공기조성 (모터에 의해 흡입되는 공기의 양이 없을 때)과 시스템과 연결된 용기내의 공기조성 (모터에 의해 흡입되는 공기의 양이 있을 때)을 GC를 이용하여 비교, 측정하였다 (그림 2-31). 실험결과 모터에 의한 공기흡입이 없을 때 저장용기내의 이산화탄소 농도는 증가하였으며, 산소농도는 감소하는 경향을 나타냈다. 이와는 대조적으로 모터에 의해 흡입되는 공기의 양이 있을 때 용기내의 가스농도는 일정하게 유지되어 모터에 의한 공기 흡입은 일종의 환기 조건 형성이라 판단되어 정확한 흡입량과 유속을 고려해야 하는 어려움이 나타났다.

## 나. 개선된 방식의 자동 측정 시스템 개발

### 1) 시스템 개요

모터에 의한 가스 흡입방식의 자동측정 시스템은 모터에 의해 흡입되는 정확한 공기의 양을 알기 어렵고 이에 따라 저장고내 기체 환경 조성의 변화가 발생하기 때문에 정확한 가스농도를 측정하기 어려운 문제점이 나타났다. 따라서 모터를 이용하여 흡입한 저장고내 가스를 저장고 밖에 위치한 가스 센서까지 운반하여 감지하는 형식이 아닌 가스 센서를 직접 저장고내에 위치시켜 농도를 감지하는 방식으로 개선하였다.

### 2) 자동측정 시스템을 이용한 저장고내 환경정보 측정 및 오차범위 최소화

자동 측정 시스템을 이용하여 마늘과 양파의 저온저장시 온·습도, 이산화탄소, 산소의 환경 정보 변화를 측정하였다. 또한 저장고의 설정 기준치와 자동 시스템 측정치간의 오차범위를 최소화하고 신뢰성 있는 측정 데이터의 확보를 위하여 저장용기내의 온·습도 및 가스 농도를 각각 자동 온·습도 기록계와 GC(TCD, FID)를 이용하여 측정한 후 동일한 시간대에 시스템 데이터와의 비교를 계속함으로써 오차 범위를 최소화하고자 하였다.

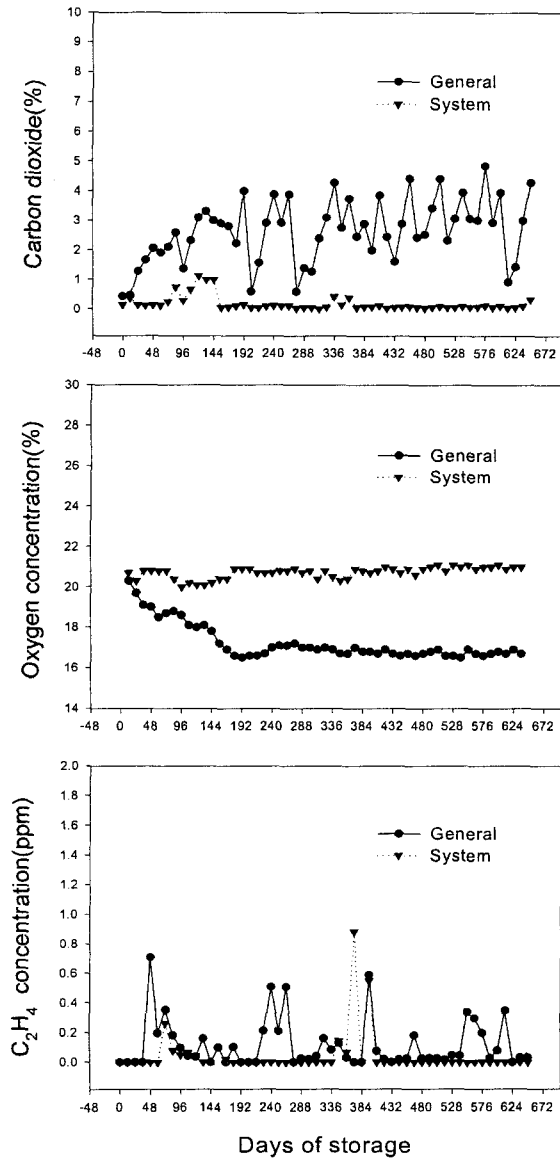


Fig. 2-31. CO<sub>2</sub>, O<sub>2</sub>, and C<sub>2</sub>H<sub>4</sub> data in garlic measured by GC  
Garlic was connected with or not connected with system.

가) 사과

그림 2-32는 후지 사과를 0°C에서 210일 동안 저장하면서 이산화탄소, 온도 및 습도 항목을 자동측정 시스템 데이터와 GC 및 자동온·습계를 이용해, 측정된 데이터를 지속적으로 비교한 것이다. 이산화탄소 농도의 경우 system과 manual의 데이터가 거의 오차가 없었다. 온도의 경우 두 데이터간의 경향은 유사했으나, 3°C 정도 차이가 있는 것으로 나타났다. 습도는 system과 manual(자동온습계)의 데이터 사이에 적게는 0.02%, 많게는 8.04% 정도의 오차를 보이는 등 비교적 불규칙한 양상을 보였다.

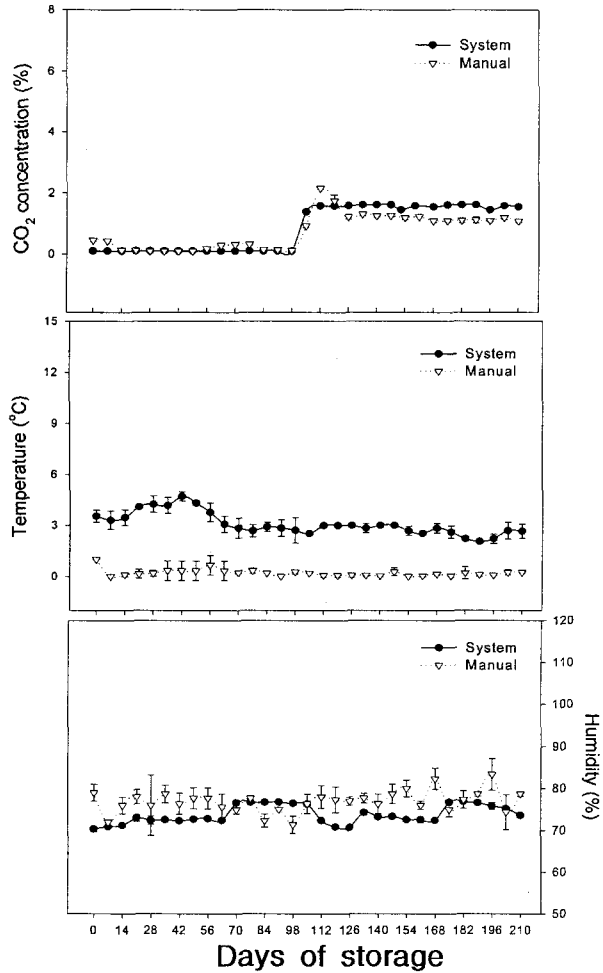


Fig. 2-31. Changes of CO<sub>2</sub> concentration, C<sub>2</sub>H<sub>4</sub> concentration, temperature and humidity within storage chamber 'Fuji' apple during storage at 0C for 210 days. Brs show standard deviation.

나) 배

그림 2-33은 신고 배를 0°C에서 135일간 저장하면서 이산화탄소, 에틸렌, 온도 및 습도 항목을 자동측정 시스템 데이터와 GC 및 자동온·습계를 이용해 측정한 데이터를 지속적으로 비교한 것이다. 이산화탄소의 경우 system과 manual의 데이터가 거의 차이가 없는 것으로 나타났다. 하지만 온도의 경우 경향치는 유사하게 나타났지만 4°C 정도의 오차를 보였다.

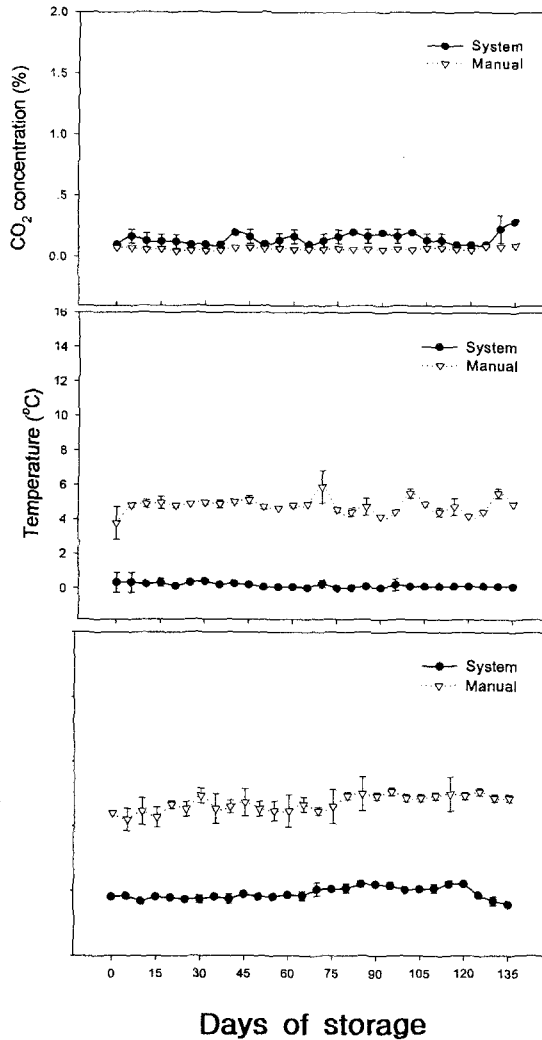


Fig. 2-33. Changes of CO<sub>2</sub> concentration, C<sub>2</sub>H<sub>4</sub> concentration, temperature and humidity within storage chamber ' Niikata' pear during storage at 0C for 135 days. Bars show standard deviation.

다) 양파

그림 2-34는 양파를 0°C에서 30일간 저장하면서 습도, 산소, 이산화탄소 항목을 자동 측정 시스템 데이터와 GC 및 자동온습계를 이용해 측정한 데이터를 지속적으로 비교한 것이다. 습도의 경우 system과 manual 데이터 사이에 많게는 18%, 적게는 10% 정도의 오차가 있었다. 산소는 두 데이터가 미세한 차이는 있지만 매우 유사한 것으로 나타났으며 이산화탄소도 마찬가지로 오차범위가 1% 미만으로 비교적 일치하는 경향이였다.

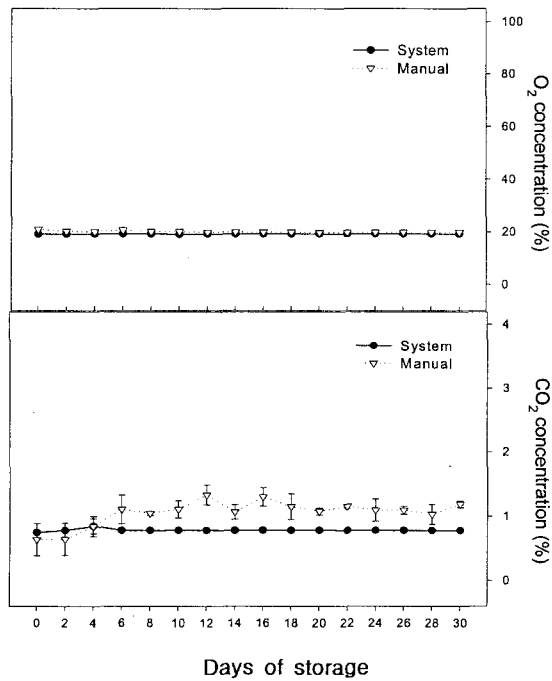


Fig. 2-34 Changes of humidity, O<sub>2</sub> concentration, CO<sub>2</sub> concentration and C<sub>2</sub>H<sub>4</sub> concentration within storage chamber of onion during storage at 0C for 30 days. Bars show standard deviation.

라) 마늘

그림 2-35는 마늘을 0°C에서 30일간 저장하면서 습도, 온도, 이산화탄소 항목을 자동측정 시스템 데이터와 GC 및 자동온·습계를 이용해 측정된 데이터를 지속적으로 비교한 것이다. 습도는 경향치는 유사했으나 상대적으로 높은 15~30% 정도의 오차범위를 나타냈다. 온도의 경우 2°C 정도의 약간의 오차는 있었지만 system과 manual 데이터가 비교적 일치하는 경향을 보였다. 이산화탄소는 0.5% 정도의 오차가 있었지만 비교적 일치하는 경향을 보였다.

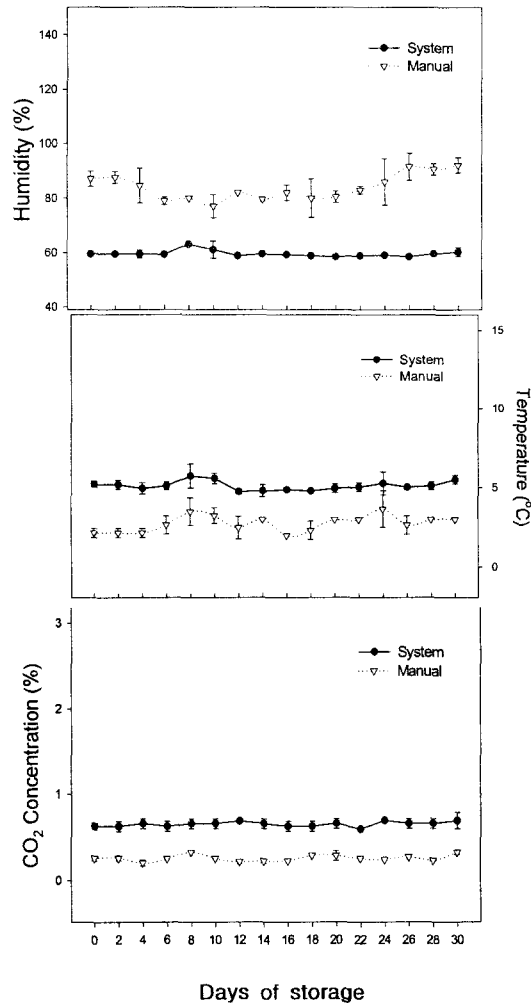


Fig. 2-35. Changes of humidity, O<sub>2</sub> concentration, CO<sub>2</sub> concentration and C<sub>2</sub>H<sub>4</sub> concentration within storage chamber of garlic during storage at 0°C for 30 days. Bars show standard deviation.

다. 상업용 저장고의 규모에 따른 환경관리의 통일된 시스템을 개발

1) 시스템 개요

지금까지 하드웨어 및 소프트웨어 개발팀에 의해 공동으로 구축된 실험실 차원의 자동 측정 시스템 장치를 상업용 저장고 규모로 확대, 적용하기 위하여 2ton 규모 상업용 pilot 저장고 2개에 각각 온도, 습도, 이산화탄소, 산소센서를 배치하고, 사과 (제 2저장고)와 배(제 1저장고)를 전반기에, 마늘(제1저장고)과 양파(제2저장고)를 후반기에 저장하였다 (그림 2-36).

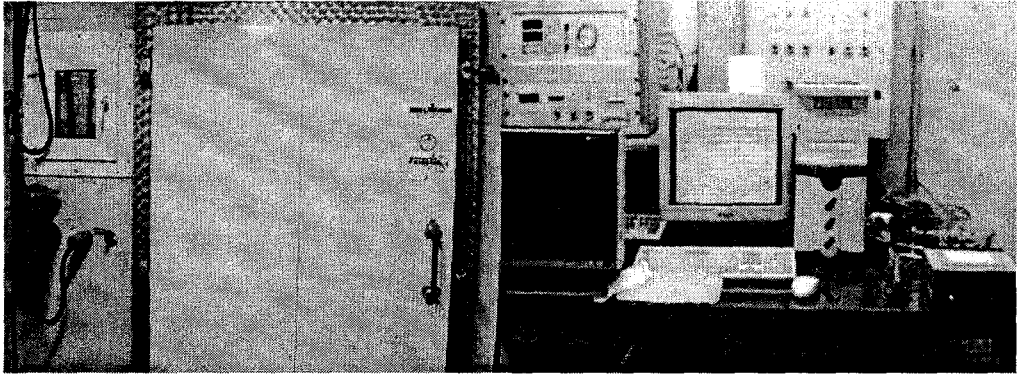


그림 2-36. 환경관리자동화 시스템 (좌) 및 시스템에 사용된 상업용 규모의 저온저장고(우)

2) 자동측정 시스템을 이용한 저장고내 환경정보 측정 및 오차범위 최소화

현재까지 상업용 저장고 규모(2ton)로 구축된 자동측정 시스템을 이용하여 저장고내의 온, 습도 및 가스변화를 측정한 data는 <http://203.237.178.103/nong2/index.html>에 누적되어 있다. 자동 측정 시스템의 오차범위를 최소화하고 신뢰성 있는 데이터의 확보를 위하여 저장고내의 온·습도를 자동 온·습도 기록계를 이용하여 측정한 후 시스템 데이터와 지속적으로 비교하여 함으로써 오차범위를 최소화 하고자 하였다. 또한 이산화탄소와 산소의 경우는 표준가스(이산화탄소 10.9%, 산소 10.0%)를 이용하여 센서에서 나오는 데이터와 표준가스 농도를 비교하여 자동측정 시스템의 오차범위를 최소화하고자 하였다.

### 제 3절 결론

현재 농가 저온저장고의 면적이 증가하는 추세임에도 불구하고 작물의 저온저장에 따른 환경조성과 저장요인의 변화를 수시로 관리, 예측하지 못하는데 따른 피해가 해마다 증가하고 있다. 따라서 본 연구에서는 저장고내 환경을 자동으로 관리하고 조절할 수 있는 시스템의 구축을 목적으로, 원예작물의 저장고내 품질과 관련된 여러 환경 요인들을 일차적으로 센서에서 감지하고 그 정보를 컴퓨터에 자동으로 입력하여 궁극적으로 환경요인을 자동으로 관리할 수 있는 시스템을 개발하였다.

본 시스템의 개발을 위해서 원예작물의 저온저장시 품목차이에 따른 수확후 생리특성을 조사하고 이에 따른 작물의 품질변화를 지속적으로 모니터링 함으로써 원예작물의 저장특성을 구명하고, 환경관리 자동화 시스템 개발에 대한 기초자료를 제공하였다.

지금까지 하드웨어 및 소프트웨어 개발팀에 의해 공동으로 구축된 자동 측정 시스템 장치를 이용하여 상업용 저장고내의 온, 습도 및 가스변화를 측정한 data는 <http://203.237.178.103/nong2/index.html>에 누적되어 있다.

궁극적으로 본 과제를 통해 개발된 저온저장고 환경관리 자동화시스템은 저온저장고내 온도, 습도, 이산화탄소, 산소의 저장 환경인자를 최적 조절 상태로 유지시킬 수 있고, 작물에 따른 특이한 수확후 생리적 숙성 및 노화과정에 대한 변화를 용이하게 예측하게 하며, 환경정보를 관리사무실이나 집에서 쉽게 관찰할 수 있는 관리상의 편리성을 제공할 수 있으리라 기대된다.

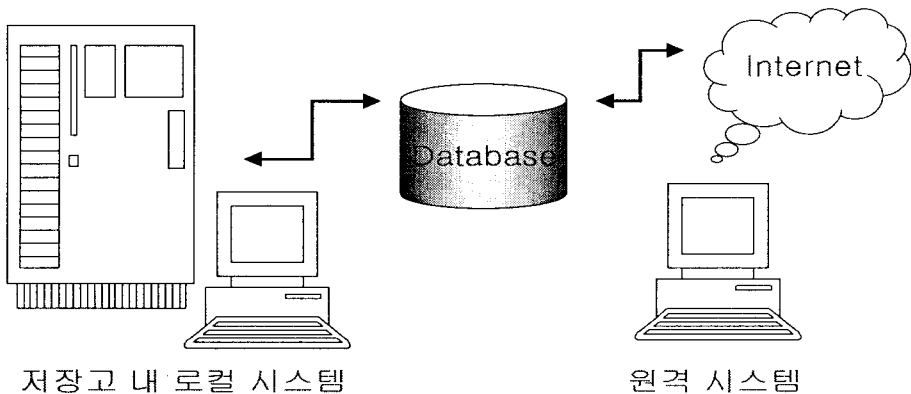


# 제3장 환경관리를 위한 자동측정 응용 소프트웨어 개발

## 제1절 서설

저온 저장 환경관리 자동화 시스템의 목적은 저장고 내의 온도, 산소, 이산화탄소, 에틸렌, 습도 등의 환경 인자들을 자동으로 측정함으로써 이를 분석하여 저장 품목의 상태를 점검하고 품목의 상품성을 향상시키는데 있다. 이를 위해서는 시스템의 제작에 앞서 자동화 시스템에 필요한 요구사항들을 정리하고 분석하는 과정이 있어야 했으며, 시스템을 구성해 나가는 동안에도 추가적인 요구사항과 문제점 등을 고려하여 더욱 효과적이고 유용하도록 프로그램을 수정, 보완하는 작업이 필요하였다. 1, 2차 년도에는 이에 맞추어 시스템 설계를 진행해왔으며, 3차 년도에는 그 시스템을 안정화시키는데 주목적을 두었다.

전체적인 구성은 크게 Visual C++로 구현한 저장고 내 로컬 시스템과 ASP(Active Sever Page)로 구현한 원격 시스템, 그리고 데이터의 보관 및 응용에 쓰이는 MS Access를 이용한 DataBase로 볼 수 있다. 전체적인 구성을 대략적으로 보면 그림 3-1과 같다.



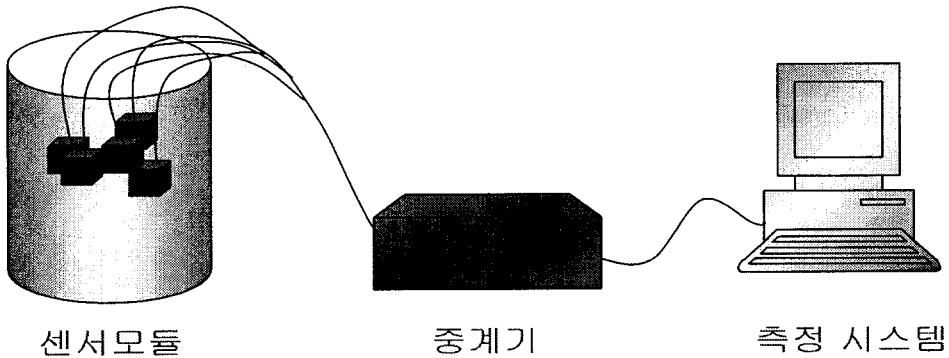
[그림 3-1] 시스템 전체 구성

다음에 설명될 내용은 측정 표준안과 기능을 고려한 메뉴별 구성을 중심으로 설명하도록 한다.

## 제2절 측정 표준안 작성

### 1. 측정 항목의 종류

저온 저장고에 저장된 작물이 최적의 상태로 유지되기 위해서는 저장 목적의 특성상 온도의 측정은 반드시 필요하며 그 외 산소, 이산화탄소, 습도, 에틸렌의 농도를 측정, 점검할 수 있도록 세부 항목에 포함하여야 한다. 각 세부 항목들은 그림 3-2에서처럼 온도, 산소, 이산화탄소, 습도, 에틸렌 각각의 센서 모듈을 통해 데이터를 측정하며 이렇게 측정된 데이터들은 중계기를 거쳐 시스템으로 그 값이 전송되게 된다.



[그림 3-2] 측정 항목 처리

### 2. 측정 데이터 정의

중계기를 통해 센서로부터 전송되는 데이터의 형태는 0 ~ 255 사이의 8비트 값이 전송된다. 이것을 각 항목에 적합한 적용 산술식을 통하여 원하는 데이터를 얻게 되는데 이때 데이터 타입은 double 형을 사용하였다. 여기서 산술식이란, 각각의 binary 수치를 항목에 맞는 유용한 데이터의 형태로 변환하기 위한 수식으로, 다음과 같은 수식으로 환

산되어 각 단위별 데이터를 얻게 된다.

```
#define TEMP (60.000000/255.000000)*float(in)-5
#define HUM (100.000000/(255.000000))*float(in)
#define CARBON1 (20.000000/255.000000)*float(in)
#define CARBON2 (25.000000/255.000000)*float(in)
#define OXY (25.000000/255.000000)*float(in)
#define ETIL (10000.000000/255.000000)*float(in)
```

실제적으로 측정항목의 단위는 온도의 경우 °C 이고, 산소, 이산화탄소, 습도는 %, 그리고 에틸렌은 ppm으로 측정값의 유효범위가 소수 둘째 자리까지 요구되기 때문에 double 형을 설정해 놓음으로써 필요할 때 더욱 세밀한 표현까지 가능하도록 한다.

### 3. 저장고 및 내부 센서 등 시스템 확장성 고려

규모에 따라 저장고나 그 내부 센서의 개수는 증가할 수 있다. 예를 들어 대형 저장고의 경우 온도센서를 하나가 아니라 여러 개를 서로 다른 위치에 배치하는 경우가 있고, 더 나아가서 그러한 대형 저장고가 여러 개인 경우도 있기 때문이다. 그러므로 센서 설치 현황에 맞게 확장성이 고려되어야 한다.

초기에는 확장성을 고려하여 그림 3-3과 같이 모터를 사용하여 각 저장고를 제어하였다. 하지만 기존의 모터 방식에 여러 가지 불편사항들과 번거로운 문제점들로 인해 이를 바꾸어 중계기를 사용하는 방법으로 변형하였다.

제 1 저장고 <주소값 : 0x00>

온도	: 0x00 ~ 0x0F
습도	: 0x10 ~ 0x1F
이산화탄소	: 0x20 ~ 0x2F
산소	: 0x30 ~ 0x3F
에틸렌	: 0x40 ~ 0x4F

.....

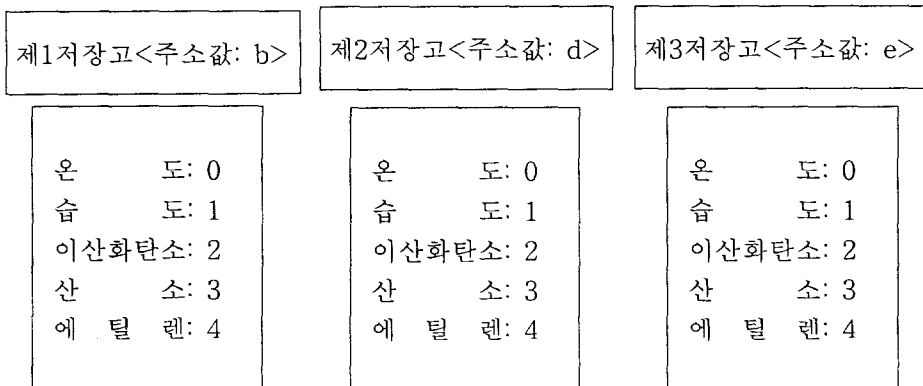
제 256 저장고 <주소값 : 0xFF>

온도	: 0x00 ~ 0x0F
습도	: 0x10 ~ 0x1F
이산화탄소	: 0x20 ~ 0x2F
산소	: 0x30 ~ 0x3F
에틸렌	: 0x40 ~ 0x4F

[그림 3-3] 초기 모터 제어 구조

실제 저장고의 형태를 보면 하나의 저장고를 이용하는 경우도 있지만 여러 개의 저장고를 관리해야 하는 경우도 있으며, 또한 측정 항목에 있어서 저온저장의 목적 특성상 단순히 온도만을 측정하는 경우도 있겠지만 그밖에 고려해야 할 여러 가지 항목들에 대해서 측정하는 경우도 있게 되는데 이 때 중계기에서 시스템의 요구대로 각 저장고로의 연결을 이행하게 되고, 저장 항목별로 주소 값을 두어 각 저장고와 항목들에 대해 확장성을 가질 수 있게 되었다.

변형된 중계기를 이용한 방식은 다음 그림 3-4와 같은 형식으로, 각 저장고별 주소 값을 b, d, e, f 등으로 주고 그에 따른 각 항목을 0, 1, 2, 3, 4로 주소값을 할당하는 형태를 갖게 된다.



[그림 3-4] 변경 후 중계기 제어 구조

이러한 주소의 형태를 가지므로 예를 들어 측정을 원하는 데이터가 제 1 저장고의 온도 항목일 경우 중계기로 b0 값을 전송하게 되면, 중계기는 이에 해당하는 제 1 저장고의 온도 센서 데이터를 요청한 시스템으로 반환하게 되는 것이다.

#### 4. 측정시간 간격 정의

상황에 따라 저장고의 환경인자들에 요구되는 측정 간격은 각기 다를 수 있다. 따라서 과수의 성격에 맞게 원하는 측정간격을 정할 수 있도록 설계해야 할 필요가 있었으며 테스트 결과, 짧게는 한시간에 한번씩 측정하는 경우부터 길게는 24시간마다 측정하는 경우가 필요하였기 때문에 그 사이의 간격을 여러 가지로 마련하여 시스템 상에서 원하는 측정간격을 결정할 수 있도록 하고 이렇게 결정된 측정 간격마다 센서제어기에 측정을 요구하는 형태로 구성하였다.

#### 5. 데이터 분석의 편리성 고려

시스템의 요구에 의해 센서에서 측정된 데이터는 중계기를 거쳐서 저장고 내의 컴퓨터 화면으로 모니터링 되는 동시에 데이터베이스에 저장된다. 이렇게 누적된 데이터는 필요에 따라 분석이 용이하도록 실시간으로 제공되어 프로그램 상에서 동적으로 볼 수 있어야 한다. 이러한 데이터의 변화에 대해 효과적인 비교, 분석이 가능하도록 하기 위해 정확한 값을 볼 수 있는 표 보기의 형태와 데이터의 변화 양상을 한눈에 파악할 수 있는 그래프 보기의 형태로 나누어 데이터 보기를 구성하였다.

##### 가. 데이터 보기

최종 측정시점으로부터 1일에서 5일까지의 데이터와 월별 데이터가 분석을 하는데 있어서 자주 사용됨을 알 수 있었기 때문에 누적 데이터 중 원하는 양만큼의 최근 데이터를 볼 수 있게 하였다. 또한 분석 시에 저장고별로 항목들의 변화를 보고자 하는 경우와 여러 개의 저장고에서 특정 항목에 대해 비교 분석하는 경우가 필요하다고 보고 저장고 별 보기, 항목별 보기로 구분하여 데이터 보기를 세분화하였다.

## 나. 그래프 보기

그래프 보기 역시 데이터 보기와 마찬가지로 원하는 저장고 내 특정 항목의 변화 양상을 한눈에 볼 수 있도록 꺾은선 그래프로 표현하였다. 초기 그래프 보기에서는 데이터 보기 간격을 사용자가 설정하여 볼 수가 없었다. 그래서 측정되어 저장된 데이터들의 수치가 여러 개의 점들로 나타나게 되는데 데이터가 많아지면 그래프 상의 점들이 선이 보이지 않을 정도로 많아져서 상태를 한눈에 파악하기 힘들다. 또한, 사용자가 관찰하는데 효율을 떨어뜨리게 되므로 데이터 간격을 사용자가 임의로 변경할 수 있도록 하여 그 간격마다 그래프 상에 데이터를 표시하도록 설계하였다.

## 6. 활성화된 저장고 설정

본 프로그램은 확장성을 고려하여 6개의 저장고까지 확장 가능하도록 설계하였다. 이 중 초기 버전에서는 하나의 저장고만 활성화되어 동작하였으나 센서 모듈이 추가되는 변화도 있었고 저장고의 활성화 여부가 변경 될 때마다 프로그램을 수정해야하는 번거로움을 덜기 위해서 가동되는 저장고가 변할 때마다 매번 프로그램 상에서 코드를 수정하지 않고도 사용자가 간단히 체크박스에 표시하는 작업만으로 측정이 필요한 저장고를 선택할 수 있도록 하였다.

## 7. 메뉴바와 툴바의 삭제

기존의 응용 프로그램 View 화면의 상단에는 팝업 메뉴바와 툴바가 있었다. 하지만 그동안 이용 현황을 보아온 결과 사용하는 일이 거의 없으며 오히려 시스템 이용 시 사용자에게 혼란을 더 야기 하게되는 것으로 생각되어 코드 상의 변경으로 불필요한 메뉴바와 툴바를 제거함으로써 좀 더 간편한 폼 디자인을 얻었다.

## 8. 지속적인 관찰을 위한 편리성 고려

저장고 내의 환경변화에 대해서 정확한 정보를 지속적으로 관찰하기 위해서는 시간과 장소에 구애받지 않고 편리하게 관찰할 수 있도록 원격 시스템의 형태를 갖추어야 했다. 이를 통하여 저장고 정보를 원격지에서도 쉽게 관찰할 수 있게 되어 편리성을 한층 높

었다.

다음절에서는 자동측정 시스템의 구성과 그 변화 내용을 프로그램 화면을 통해 메뉴별로 설명하도록 하겠다.

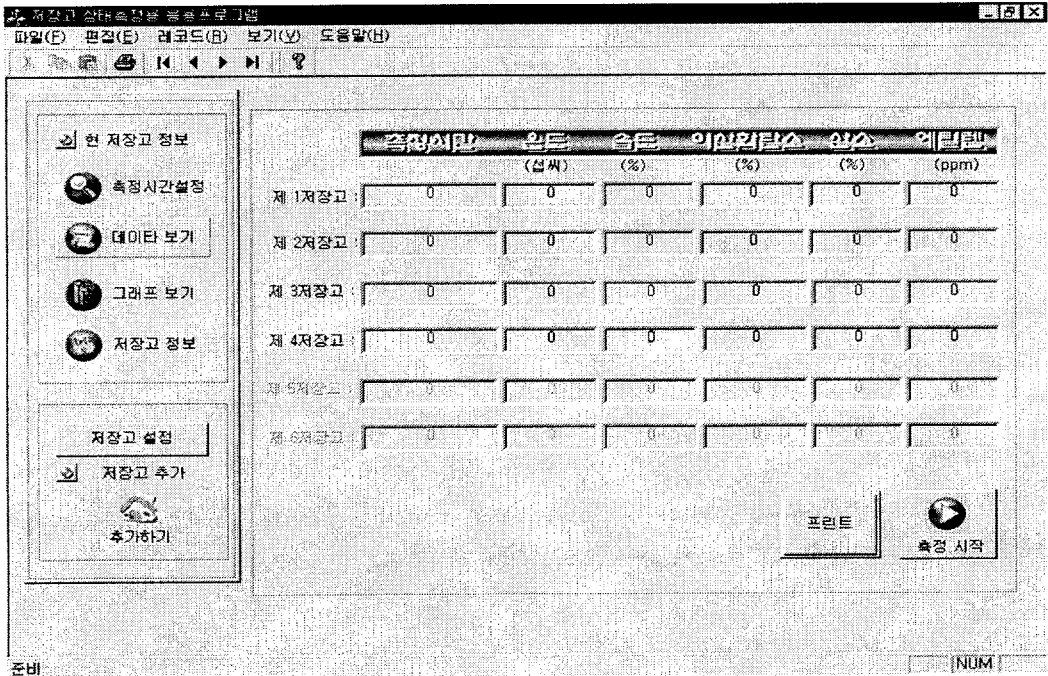
## 제3절 기능을 고려한 각 메뉴별 구성

### 1. 로컬 시스템

로컬 시스템은 Visual C++ 로 구현하였다. 이는 센서와 데이터를 주고받을 수 있는 시리얼 통신(Serial Communication)이 가능하도록 하며, MS ACCESS 데이터베이스를 구축하여 데이터의 분석, 관리가 용이하도록 하였다.

로컬 시스템에서는 사용자의 요청에 의해 중계기를 거쳐 시리얼 케이블(Serial Cable)을 통해 센서로부터 데이터가 전송되는데 그 데이터는 실시간으로 저장고 내의 환경변화에 따른 데이터의 값이 컴퓨터 화면에 나타나고 또 그와 동시에 모니터링된 데이터는 데이터베이스에 저장되는 것이다. 그리고 첫 번째 번지의 주소 값을 요청하게 되면 프로그램 설계대로 환경변화 측정을 위한 시간간격을 시스템에서 자동으로 입력받게 된다. 그렇게 함으로써 데이터의 측정시점을 정확하게 기록할 수 있게 한다.

그림 3-5는 측정 시스템의 메인 화면으로 저장고 내에 장착된 센서에서 측정된 값이 항목별로 변환 수식에 따라 실시간으로 측정되어 저장고 내의 환경 변화가 모니터링 되는 화면이다.



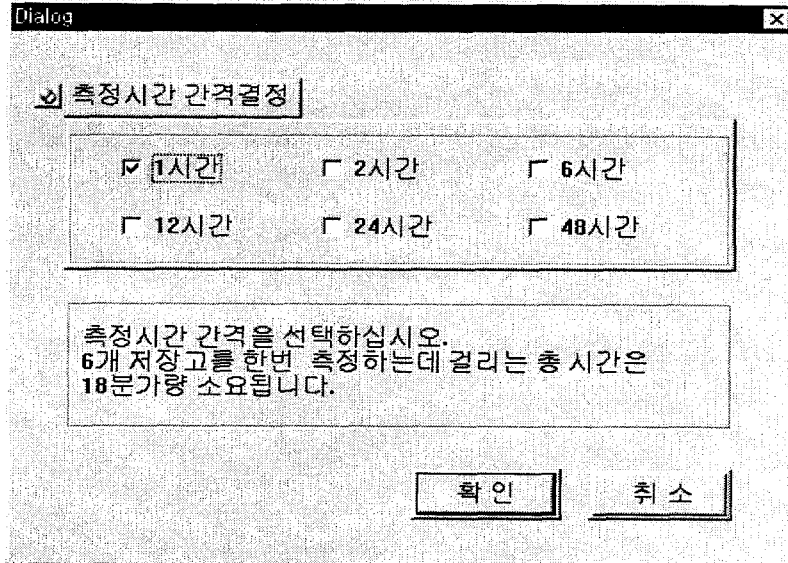
[그림 3-5] 모니터링 메인 화면

왼쪽의 메뉴부분은 측정시간 설정, 데이터 보기, 그래프 보기, 저장고 정보 및 저장고 설정과 같은 버튼들로 구성된다. 각각의 메뉴별로 설명은 다음과 같다.

#### 가. 측정시간 간격 설정

메인 화면에서 측정시간 설정 버튼을 누르면 그림 3-6과 같은 화면이 나타난다. 요구 분석의 결과에 따라 1시간에서 2, 4, 12, 24, 48 시간 등의 간격이 적당하다고 생각되어 아래 그림 3-6과 같이 다양한 간격을 설정해두어 사용자가 저장고 내 환경 변화의 측정 시간을 원하는 바에 따라 편리하게 변경할 수 있게 하였다. 한 개의 저장고 내 환경변화 항목을 측정하는 데는 대략 2분 정도의 시간이 걸리며, 따라서 확장 시켜놓은 6개의 저장고를 모두 측정하는 데는 10분 가량의 시간이 소요되게 된다.

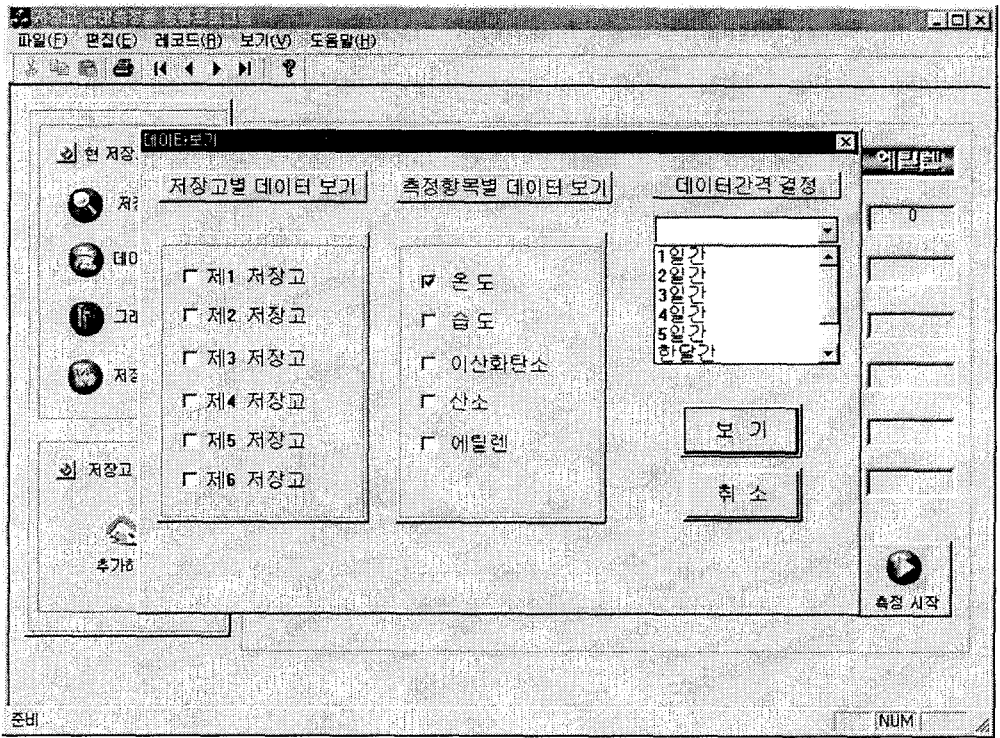




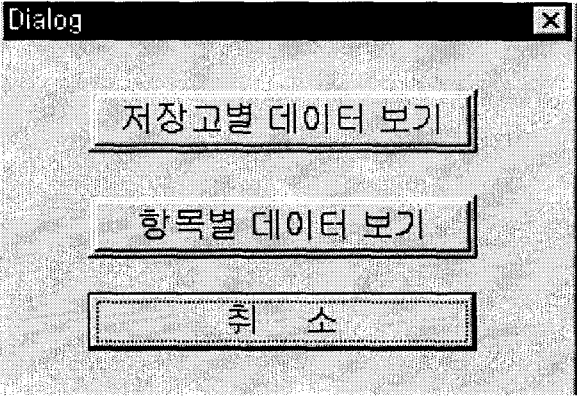
[그림 3-6] 측정시간 간격 설정

#### 나. 데이터 보기

초기 시스템의 데이터 보기는 그림 3-7과 같이 저장고별 데이터 보기와 항목별 데이터 보기가 동일한 창에 있었으므로 사용자에게 혼란을 줄뿐만 아니라 저장고별 데이터 보기와 항목별 데이터 보기를 동시에 선택할 경우 실행 오류가 나는 등 문제가 생기게 되어 프로그램의 수정이 필요했다. 그러므로 그림 3-8과 같이 저장고별 데이터 보기와 항목별 데이터 보기 버튼을 달리 하여 사용자의 혼돈을 피할 수 있었다.



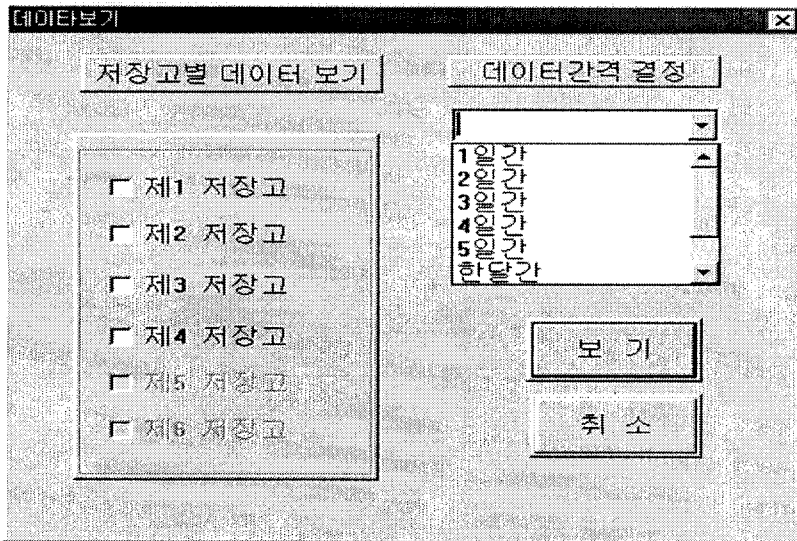
[그림 3-7] 초기 데이터 보기 대화상자



[그림 3-8] 데이터 보기

1) 저장고별 데이터 보기

먼저 저장고별 데이터 보기에서는 저장고를 선택하고 데이터 간격을 결정하면 설정 조건에 맞는 데이터를 리스트 형태로 보여준다. 데이터의 간격은 앞서서도 언급했듯이 그림 3-9처럼 저장고 내의 환경변화를 분석하는데 자주 사용되는 1일간부터 5일간까지 그리고 한 달간, 전체 보기로 나누어져 있다.



[그림 3-9] 저장고별 데이터 보기 설정

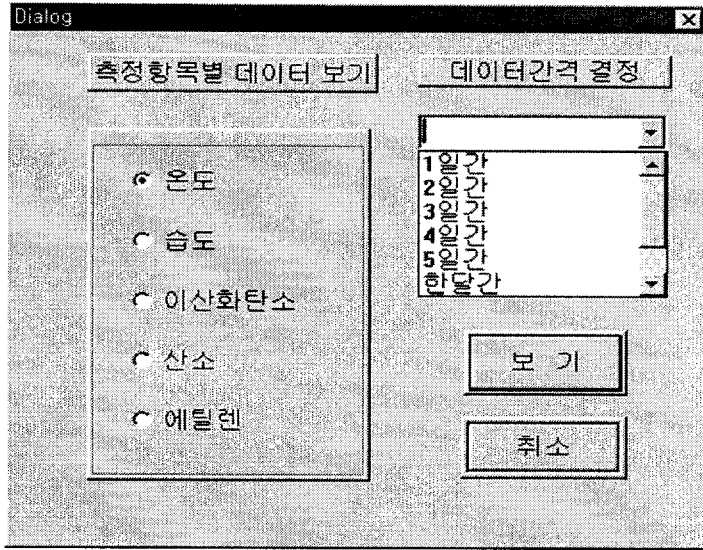
그림 3-10은 제 2저장고의 1일간 데이터보기를 한 결과 화면이다. 저장고 내 환경변화에 따른 온도, 습도, 이산화탄소, 산소, 에틸렌의 1일간 데이터들이 측정된 시간들과 함께 차례로 보여짐을 알 수 있다.

현재 시간	온도	습도	이산화탄소	산소	에틸렌
06/29/99 12:40:36	6.22	79.22	1647.06	25.00	1268.29
06/29/99 13:45:18	7.24	84.31	1725.49	25.00	1268.29
06/29/99 14:44:00	6.47	84.31	1490.20	25.00	1463.41
06/29/99 15:41:41	5.20	78.43	1647.06	25.00	1463.41
06/29/99 16:39:23	6.98	83.53	1568.63	25.00	1317.07
06/29/99 17:37:04	6.73	81.18	1647.06	25.00	1268.29
06/29/99 18:34:46	8.76	84.31	1647.06	25.00	1365.85
06/29/99 19:32:27	8.51	74.12	1568.63	25.00	1463.41
06/29/99 20:30:09	6.98	81.18	1568.63	25.00	1414.63
06/29/99 21:27:50	17.69	89.41	1568.63	25.00	1365.85
06/29/99 22:25:32	5.96	80.00	1647.06	25.00	1414.63
06/29/99 23:23:14	6.98	75.29	1647.06	25.00	1268.29
06/30/99 00:20:55	5.71	81.96	1490.20	25.00	1317.07
06/30/99 01:18:37	5.96	72.94	1490.20	25.00	1219.51
06/30/99 02:16:18	5.45	81.18	1568.63	25.00	1219.51
06/30/99 03:14:00	6.47	71.76	1803.92	25.00	1170.73
06/30/99 04:11:41	5.71	78.43	1647.06	25.00	1414.63
06/30/99 05:09:23	7.75	86.27	1960.78	25.00	1560.98
06/30/99 06:07:04	5.45	76.86	1882.35	25.00	1365.85

[그림 3-10] 저장고별 데이터 보기 결과

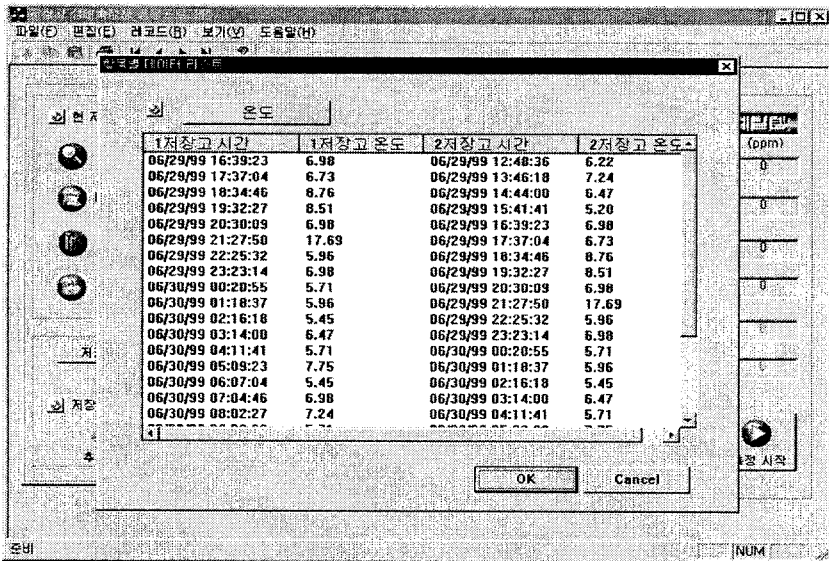
## 2) 항목별 데이터 보기

저장고별 환경 변화를 파악을 위한 데이터 보기와 유사하게 원하는 항목을 선택하고 데이터 간격을 그림 3-11과 같이 설정한다. 그러면 설정 조건에 맞는 데이터를 리스트로 보여준다. 이때 데이터 간격은 저장고별 데이터 보기와 같이 1일간에서 5일간, 한 달간, 전체 보기로 나누어진다.



[그림 3-11] 항목별 데이터 보기 설정

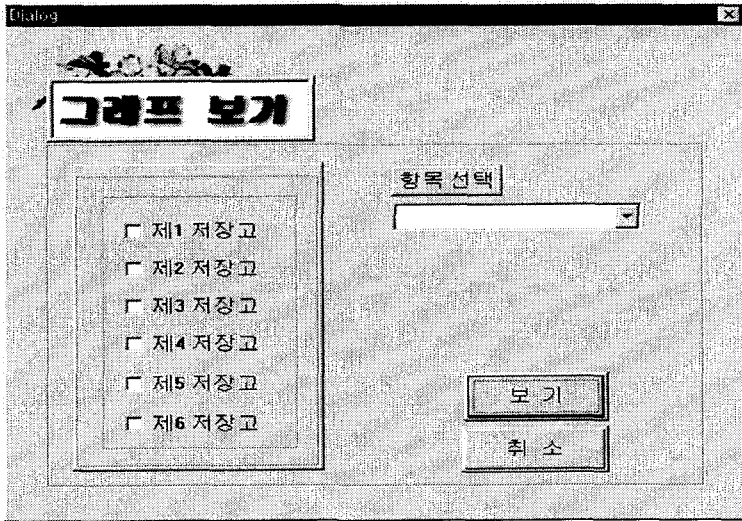
그림 3-12는 온도 항목의 1일간 데이터 보기 화면이다. 현재 가동 중인 제 1 저장고와 제 2저장고의 24시간 이내의 온도 데이터가 나란히 표시된 것을 볼 수 있다.



[그림 3-12] 항목별 데이터 보기 결과

다. 그래프 보기

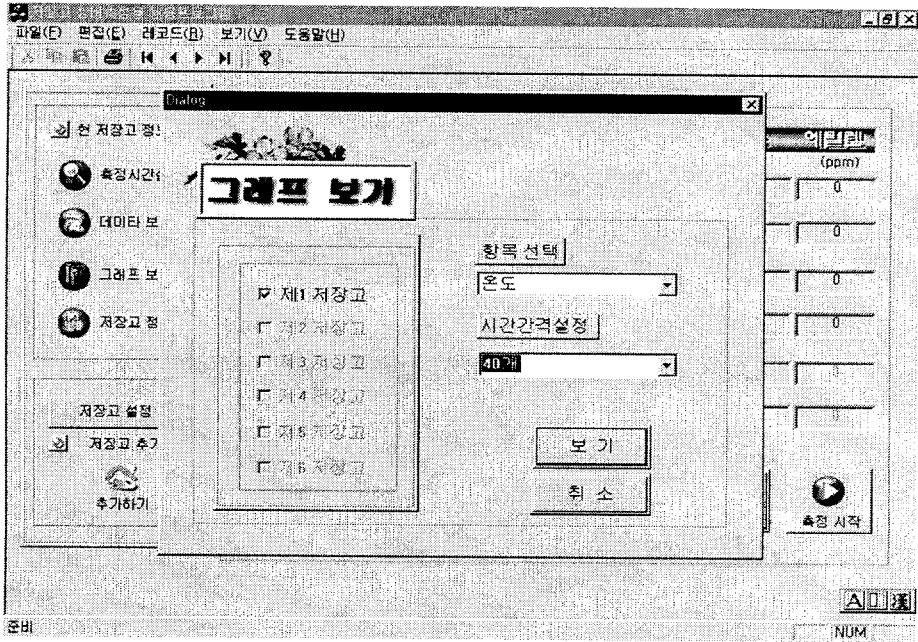
초기 시스템의 그래프 보기에서는 그림 3-13에서와 같이 저장고 내 환경변화 데이터 보기 간격을 사용자가 설정하여 볼 수가 없었다.



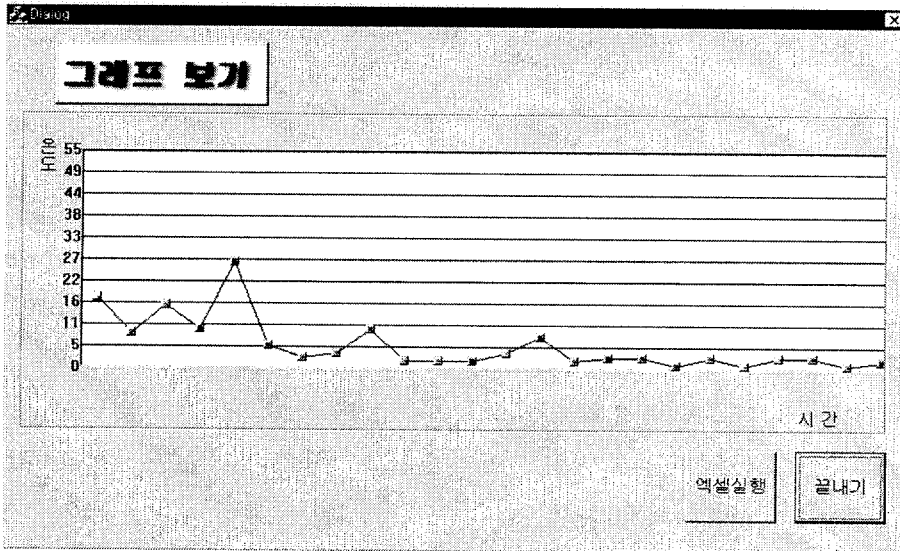
[그림 3-13] 초기 그래프 보기 설정

그래서 데이터의 양이 많아지면 그래프 상의 점들이 선이 보이지 않을 정도로 많아져서 상태를 한 눈에 파악하기 힘들뿐만 아니라 사용자가 관찰하는데 효율성을 떨어뜨리게 되므로 일정한 간격마다의 데이터를 선택할 수 있도록 재 설계하였다.

그림 3-14가 재 설계된 설정 창이며, 그림 3-15는 제 1저장고의 온도항목을 40개 데이터마다 보기 위한 결과이다.



[그림 3-14] 재 설계한 그래프 보기 설정 창

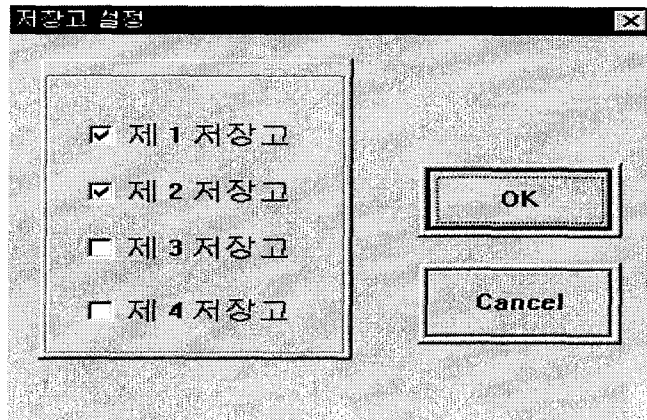


[그림 3-15] 그래프 보기 결과(제 1저장고 내 온도)

## 라. 저장고 설정

기존에는 저장고의 가동 상태가 변경 될 때마다 프로그램 상의 수정이 필요하였으며 많은 번거로움이 따랐다. 초기에 하나의 저장고만 동작하였으나 센서 모듈이 하나 더 추가되면서 2개의 저장고에서 측정이 필요하였고, 이를 통해 이후로는 저장고의 상태가 변경될 때 매번 프로그램 수정을 하지 않고도 사용자가 응용시스템 상에서 간단히 체크박스에 표시하는 것으로 같은 작업이 가능하도록 하였다.

그림 3-16은 바로 이것을 가능하게 만든 다이얼로그 상자이다.



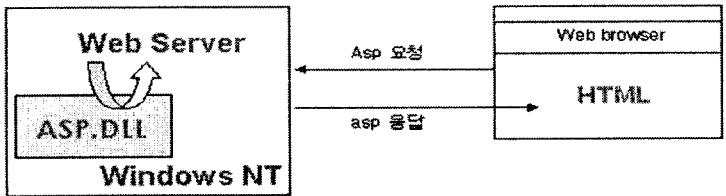
[그림 3-16] 동작중인 저장고 설정 창

## 2. 원격 시스템

Visual C++로 만들어진 로컬 시스템은 데이터를 관리하는데 여러 가지 제약이 따랐다. 그래서 시간과 장소에 구애받지 않고 저장고 내의 환경 변화에 대해서 정확한 정보를 지속적으로 관찰하기 위해서는 원격 시스템의 형태를 갖출 필요가 있었다. 이를 위해서 Visual 프로그램의 내용의 일부를 웹기반 프로그램으로의 변경을 시도하였으며 이에 따라 웹 상의 홈페이지 내에서 실시간으로 데이터를 보고 그래프 분석까지 할 수 있도록 구현되었다. 이를 통해 원격지에서도 편리하게 데이터를 관찰, 분석할 수 있게 되었다.



웹기반의 시스템을 위해서는 ASP를 이용하였다. ASP란, Active Server Pages의 약자로 IIS(Internet Information Server)의 세 번째 버전인 Microsoft사 NT의 IIS 3.0 이상에서만 동작하는 특별한 페이지이다. 이는 CGI의 과도한 서버 부하를 줄이면서도 HTML의 정적인 단점을 보완한 동적인 페이지를 나타낸다.



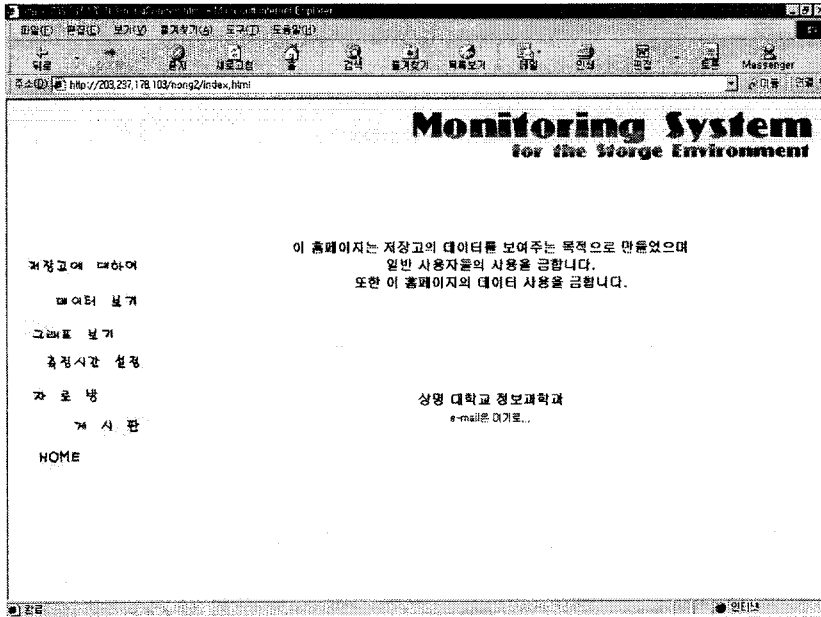
[그림 3-17] ASP의 작동 원리

위의 그림 3-17은 클라이언트가 서버에 정보를 요청할 경우 동작 원리를 보여주고 있다. HTML의 한계를 대신해서 등장한 ASP는 서버와 연동하는 작업이 가능한 특별한 페이지로 존재한다. 반드시 서버를 거쳐서 1차 해석이 되어지고, 그 결과로 만들어진 HTML을 사용자에게 보내게 한다. 클라이언트가 요청한 정보는 무조건 ASP.DLL을 거치며, 이를 통해 해석이 되어진 후에야 사용자에게 전송되어지는 것이다.

때문에 사용자는 매번 update되는 정보를 얻을 수는 있지만 단순한 HTML 형태만을 볼 수 있으며 서버의 소스는 알 수 없게 되므로 보안의 문제를 해결할 수 있다.

이 원격 시스템은 Visual C++로 된 로컬 시스템에서 모니터링한 저장고 내의 환경변화 데이터를 데이터베이스에 저장하게 되고, 이렇게 DB에 저장된 데이터를 읽어와서 웹 페이지로 변환하게 된다.

그림 3-18은 홈페이지의 메인 화면으로 인터넷 익스플로러(Internet Explorer) 4.0 버전에서 최상으로 보일 수 있도록 설계되었다.



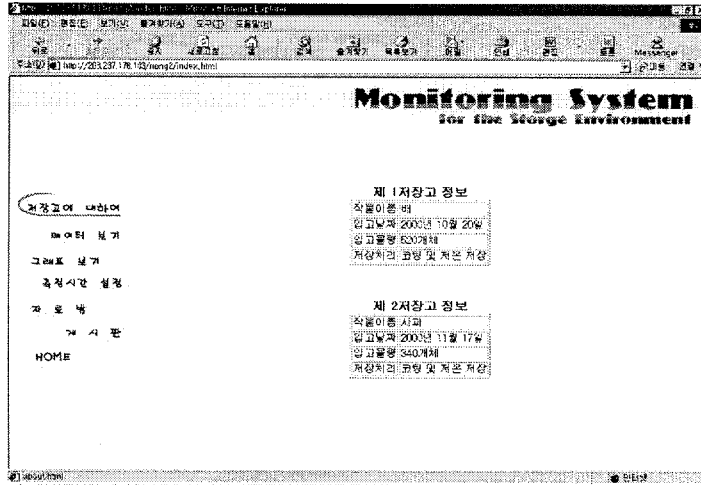
[그림 3-18] 홈페이지 메인 화면

왼쪽에 보이는 저장고에 대하여, 데이터 보기, 그래프 보기, 측정시간 설정, 자료방, 게시판, HOME 등의 메뉴가 있는데 차례대로 살펴보도록 한다.

가. 저장고에 대하여

각 저장고별 작물의 이름, 입고 날짜, 입고 물량, 저장 처리 방식 등이 종종 변경될 수 있으므로 이를 적어두어 현재 저장고 내 작물의 정보를 볼 수 있도록 하였다.

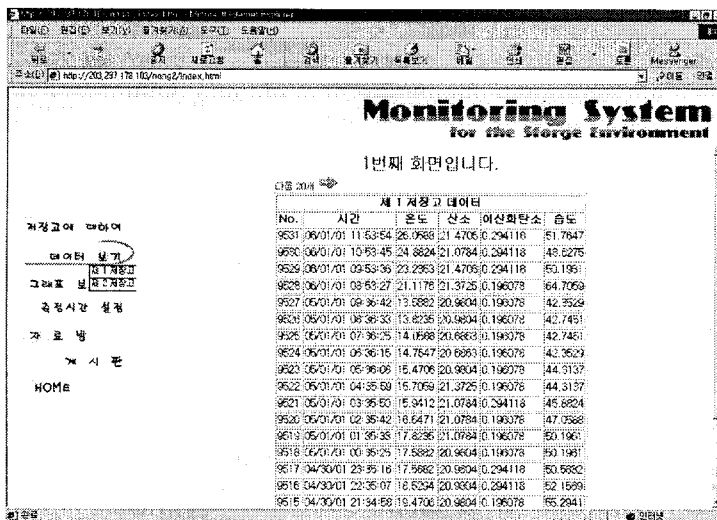
그림 3-19는 현재 저장고 내 정보를 나타내는 화면이다.



[그림 3-19] 저장고에 대하여

나. 데이터 보기

그림 3-20은 제 1저장고의 데이터 보기를 선택했을 경우의 화면이다. 저장고 내 온도 산소, 이산화탄소, 습도, 에틸렌 등의 환경 인자들이 모두 표시되는 것을 볼 수 있다. 처음에 최근 20개의 데이터 레코드가 보여지며, 화면의 상단에 있는 [다음 20개] 아이콘을 눌러서 보여지는 레코드를 이동하도록 하였다.



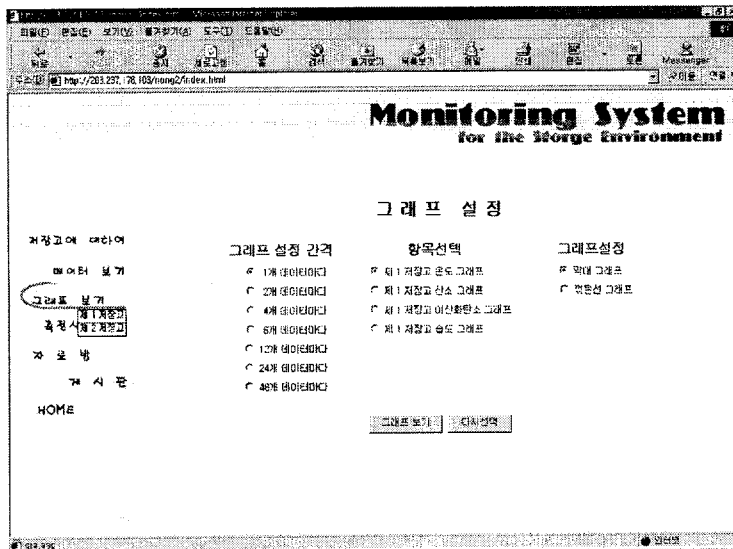
[그림 3-20] 데이터 보기

다. 그래프 보기

그래프 보기에서는 먼저 설정 페이지로 넘어간다. 여기서는 어떤 항목의 데이터를 볼 것인가를 결정하고 그려질 그래프 데이터의 간격을 설정해준다. 그래프 간격 설정은 장기간 동안의 데이터를 볼 때 데이터의 전체 흐름을 잘 파악할 수 없으므로 1개나 2개, 4개 등의 데이터마다 하나의 데이터를 추출하여 전체적인 흐름을 파악하도록 하였다.

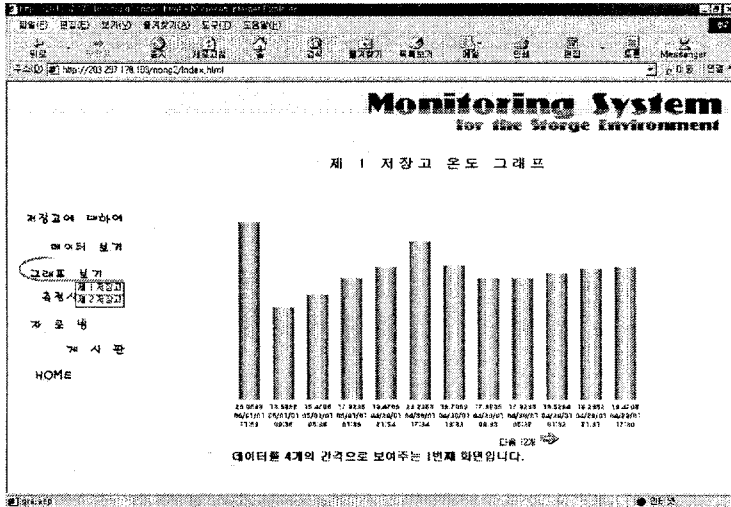
현재는 막대 그래프와 꺾은선 그래프. 두 가지의 종류가 있는데 그 중에서 꺾은선 그래프를 주로 사용하는 것으로 보인다.

그림 3-21은 그래프 보기를 원하는 저장고와 그 안의 세부 항목을 선택하고 그래프의 모양을 결정하기 위한 설정 화면이다.



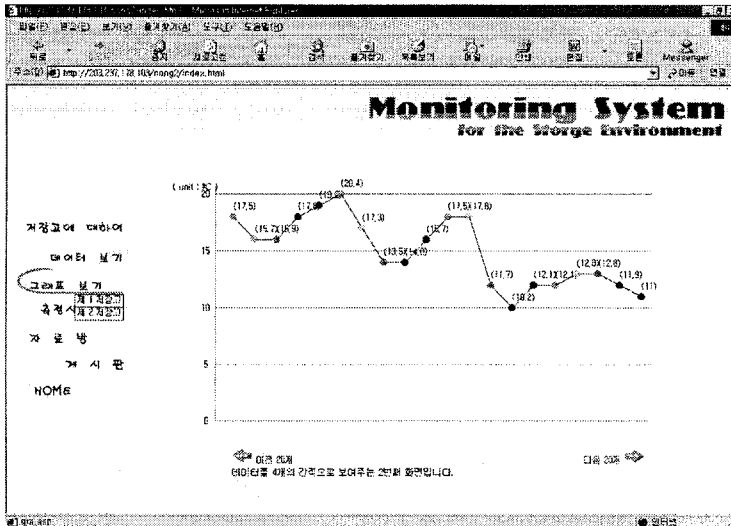
[그림 3-21] 그래프 보기 설정 창

그림 3-22는 제 1저장고의 온도를 4개의 환경변화 데이터마다 표시하도록 설정한 결과의 막대 그래프이다. 막대 그래프는 그 폭이 넓으므로 한번에 12개씩의 측정 데이터를 보여준다.



[그림 3-22] 막대 그래프 보기

그림 3-23은 위의 막대 그래프 보기와 같이 제 1저장고의 온도를 4개의 환경변화 데이터마다 표시하도록 설정된 꺾은선 그래프이다. 꺾은선 그래프는 점과 선의 간단한 구조이므로 막대 그래프 보다는 훨씬 많은 데이터를 한 화면에 표시할 수 있기 때문에 20개씩의 측정 데이터를 표현하도록 하였다.



[그림 3-23] 꺾은선 그래프 보기

라. 측정 시간 설정

초기에 로컬 시스템인 Visual C++ 프로그램 상에서만 이루어졌던 측정시간 간격설정 기능을 웹페이지에 추가하여 원격지에서도 편리하게 저장고 내의 환경 변화 데이터 관리를 위한 환경을 설정하며 필요에 따라 측정시간 간격을 자유롭게 설정할 수 있도록 하였다.

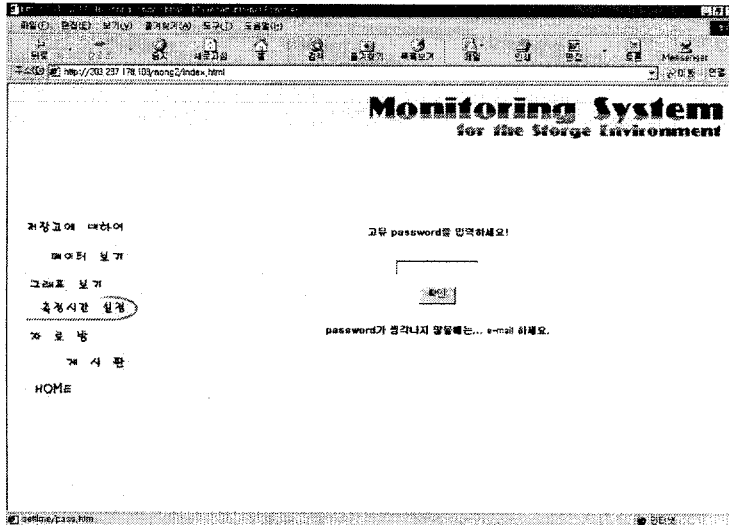
아래의 그림 3-24는 Visual C++ 프로그램과 웹페이지를 연결하여 측정간격을 설정할 수 있는 시스템의 구성을 보여준다.



[그림 3-24] 시스템의 측정시간 간격 설정 구성

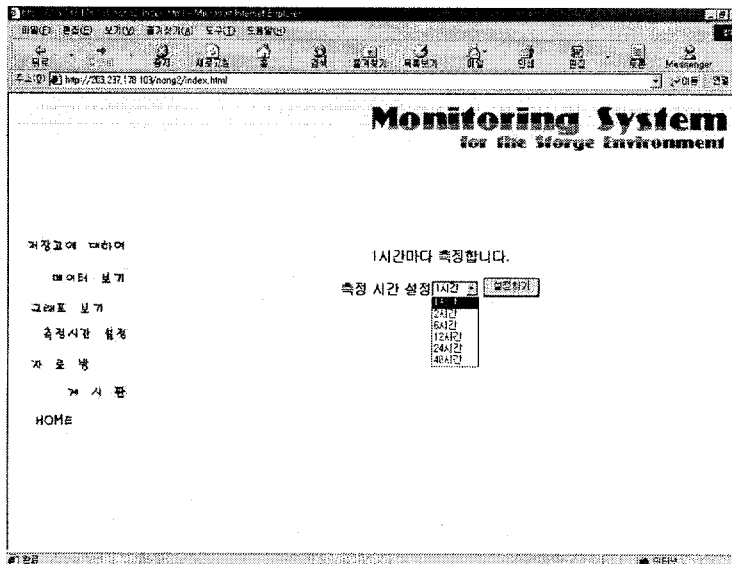
데이터베이스에 시간 변수를 두고, Visual C++ 프로그램과 웹페이지에서 모두 시간 변수에 액세스할 수 있도록 하며, 센서 값은 데이터베이스의 시간 변수에 저장된 값의 간격마다 데이터를 측정하게 되어 로컬 시스템과 원격 시스템에서 모두 측정시간 간격을 설정할 수 있게 하였다.

그러나, 이 시스템의 중요한 설정 사항을 모든 사용자가 접근할 수 있게 되면 여러 가지 혼란이 야기될 수 있으므로 시간 설정을 변경하기에 앞서 사용 권한을 확인할 수 있도록 비밀번호를 점검하여 권한이 주어진 사용자만이 접근할 수 있게 하여 저장고 내의 환경변화 측정 시스템의 보안문제를 해결하였다. 그림 3-25가 그 화면이다.



[그림 3-25] 측정 시간 변경 권한 확인

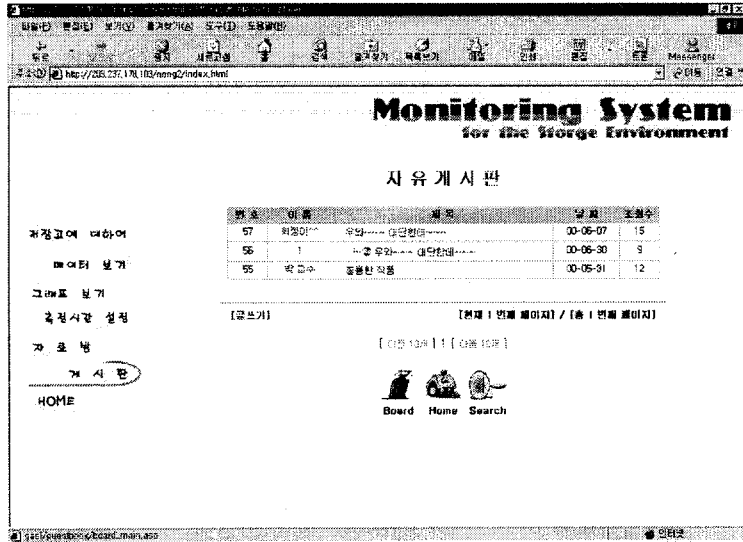
그림 3-26은 웹페이지 상에서 측정 시간 간격 설정을 하기 위한 화면이다. 설정 가능한 측정시간은 로컬 시스템인 Visual C++ 프로그램에서처럼 1시간, 2시간, 6시간, 12시간, 24시간, 48시간 등으로 분류하여 통일성을 주었다.



[그림 3-26] 측정시간 간격 설정

마. 자료방 및 게시판

홈페이지의 기본 요소인 자료방과 게시판도 추가하여 필요시에 이용할 수 있도록 하였다. 그림 3-27은 게시판의 화면이다. 화면을 전체적으로 녹색을 사용하여 구성하였기 때문에 저장고의 이미지도 살리고 간결한 디자인을 얻었다.



[그림 3-27] 게시판

## 제4절 결론

본 연구에서는 원예작물의 저온저장 중 보다 효율적인 환경관리를 위한 응용 소프트웨어 시스템을 구축하였으며 시간과 장소에 구애받지 않고 사용할 수 있도록 원격 시스템도 가능하도록 하였다. 저장고 내의 온도, 산소, 이산화탄소, 습도 등의 환경 인자들을 실시간 자동으로 측정하여 이를 분석하고 저장 품목의 상태를 점검하여 품목의 상품성을 유지하도록 편리성을 제공한다. 원격 시스템은 웹 페이지를 이용함으로써 인터넷상에서 환경변화를 측정하는 센서에서 전송하는 데이터들을 볼 수 있도록 함으로 작물에 대한



변화를 용이하게 관찰할 수 있다. 또한 인터넷을 이용하여 원격지에서도 쉽게 사용할 수 있으므로 거리에 제한을 두지 않는 관리를 위한 편리성을 제공한다.

그러므로 원예작물의 저장상태 변화에 따른 피해를 최소화하여 작물의 경쟁력을 강화 시킴으로서 추가적인 농가소득을 가져다 줄 수 있다.

# 환경관리 시스템용 센서제어기 및 모니터링 하드웨어 개발

2001. 10. 11.

협동연구기관명 : 한양대학교

협동연구책임자 : 최 명 렬

연 구 원 : 이 상 민

연 구 원 : 권 재 우

연 구 원 : 박 재 현

연 구 원 : 황 희 찬

연 구 원 : 남 상 섭

연 구 원 : 김 중 목

## 제4장 환경관리 시스템용 센서제어기 및 모니터링 하드웨어 개발

### 제1절 서설

환경관리 시스템용 센서제어기 및 모니터링 시스템은 센서부, 중계기, 호스트 PC로 구성되어 있다. 센서부는 일정시간마다 측정된 데이터를 저장하고, 중계기로 전송하는 기능을 수행한다. 중계기는 4개까지 센서부 연결이 가능하다. 그림 4-1은 전체 시스템 블록도를 도시하였다.

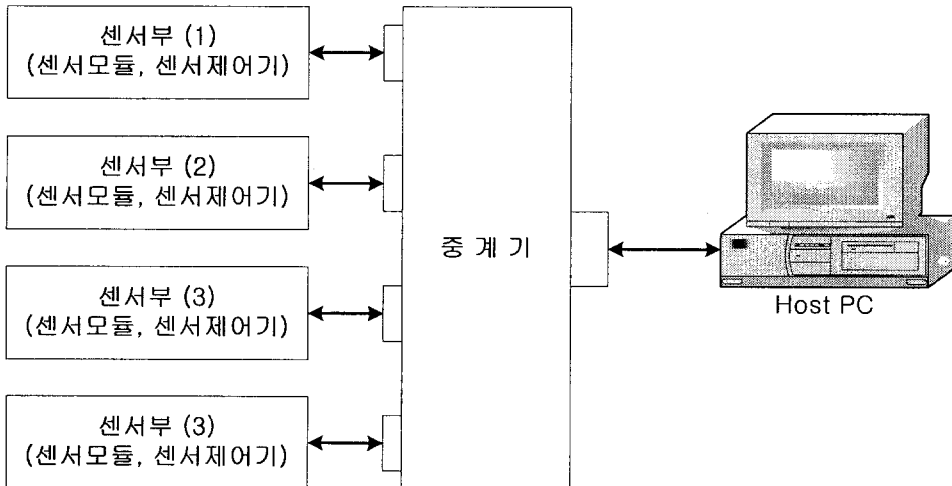


그림4-1. 시스템 블록도

## 제2절 센서부의 구성

본 과제에 센서 모듈부는 C<sub>2</sub>H<sub>4</sub>(에틸렌), O<sub>2</sub>(산소), CO<sub>2</sub>(이산화탄소), 온/습도의 네 부분으로 구성된다. 그 구성도를 그림4-2에 나타내었다.

각 센서에 입력되는 전원은 C<sub>2</sub>H<sub>4</sub>(에틸렌), O<sub>2</sub>(산소), CO<sub>2</sub>(이산화탄소)의 경우, DC 24V가 입력되어 출력 전류 4~20mA를 출력하게된다. 온/습도 센서의 경우, DC -15V가 입력되어 출력 전류 4~20mA를 출력하게된다. 표4-1은 본 과제에서 사용되는 센서의 범위 및 오차를 요약하였고, 표4-2는 각 센서에 입력되는 전압 및 출력 전류를 나타내었다.

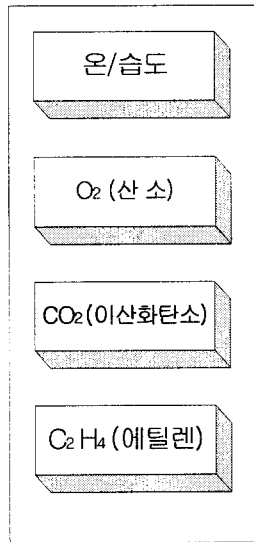


그림4-2. 센서모듈 구성도

표4-1. 센서 사양

센서 종류	허용 오차	검출 범위
온도	±3% RH( ±0.12 ℃ )	-5 ~ +55℃
습도	±3% RH	0 ~ 100 %
O <sub>2</sub>	±0.1 %	0 ~ 30 %
CO <sub>2</sub>	±0.1%FS, 0.15% of reading	0 ~ 20 %
C <sub>2</sub> H <sub>4</sub>	±1 ppm	0~10,000 ppm

표4-2. 센서 입력 전압 및 출력 전류

센서 종류	입력 전압	출력 전류
온/습도	DC -15V	4~20mA
O <sub>2</sub>	DC 24V	4~20mA
CO <sub>2</sub>	DC 24V	4~20mA
C <sub>2</sub> H <sub>4</sub>	DC 24V	4~20mA

1. 온/습도 센서

본 과제에서 사용한 온/습도 센서는 G&V사의 GHW-20(R/T)라는 모델을 사용하였다. 본 센서 허용오차는  $\pm 3\%$ 이고, 입력 공급 전압은 -15V이다. 측정 범위는 온도 센서의 경우  $-5 \sim +55^{\circ}\text{C}$ 이고 습도의 경우는  $0 \sim 100\%$ 이다. 그림4-3은 입/출력 연결도를 나타내었고, 그림4-4는 온/습도 센서의 외형을 나타내었다.

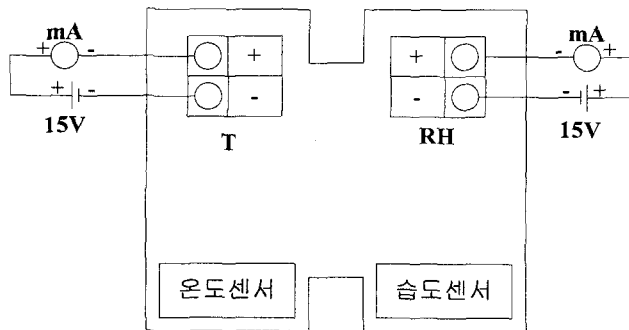


그림4-3. 입/출력 연결도

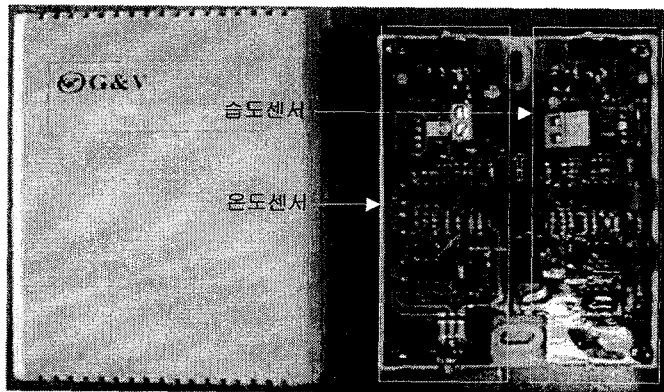


그림4-4. 온/습도 센서의 외형

## 2. O<sub>2</sub>(산소) 센서

본 과제에서 사용한 산소 센서는 허용 오차 값이  $\pm 0.1\%$  이고, 입력 전압은 DC 10 ~ 35 V 범위 내에서 자유롭게 선택하여 사용할 수 있다. 그러나 일반적으로 하나 이상의 센서에 연결되는 기록계의 부하가 큰 경우에는 DC 24V 이상을 입력 전압으로 사용하는 것이 좋다. 측정 범위는 0 ~ 25% 일 때, 출력 신호는 DC 4 ~ 20mA이다. 그림4-5는 산소 센서의 외형을 나타내었다.

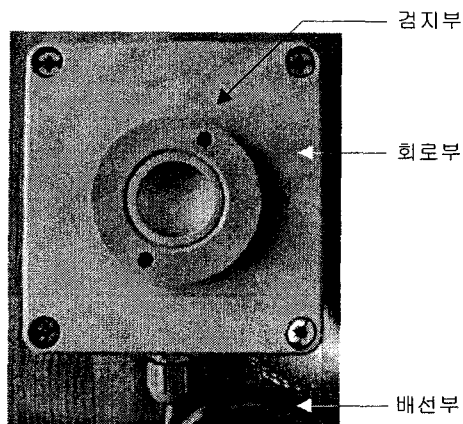


그림4-5. O<sub>2</sub>(산소) 센서

### 3. CO<sub>2</sub>(이산화탄소) 센서

본 과제에서 사용한 센서는 VAISALA사의 제품의 GMT221를 사용하였다. 공급 전압은 24 VAC/DC 두 가지 모두 사용 가능하다. 출력 데이터 값은 사용자 목적에 부합하도록 전압 및 전류 중 선택이 가능하다. 본 과제에서는 입력을 DC 24V를 사용하였고, 출력은 전류로 4 ~ 20mA까지 측정할 수 있다. 센서의 검출 범위는 0 ~ 20% 이고 허용 오차는  $\pm 0.1\%FS$ , 0.15% 이다. 그림4-6은 이산화탄소 센서의 결선도를 나타내었고, 그림4-7은 CO<sub>2</sub>(이산화탄소) 센서의 외형이다.

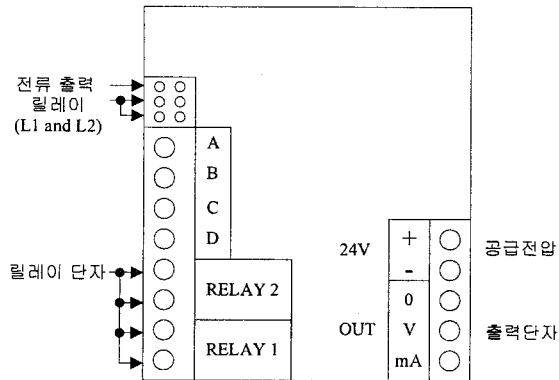


그림4-6. CO<sub>2</sub>(이산화탄소) 센서 결선도

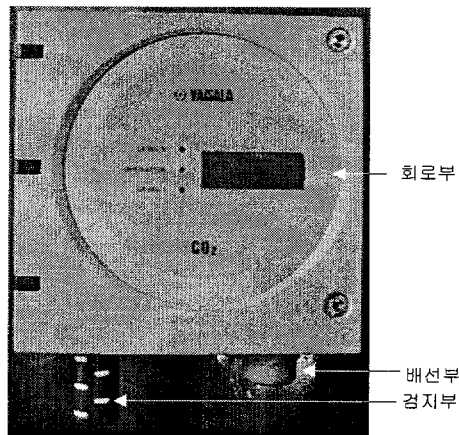


그림4-7. CO<sub>2</sub>(이산화탄소) 센서

#### 4. C<sub>2</sub>H<sub>4</sub>(에틸렌) 센서

본 과제에 사용한 C<sub>2</sub>H<sub>4</sub>(에틸렌) 센서에 공급되는 전압은 DC 10 ~ 35V로 사용자가 선택이 가능하며, 본 과제에서는 범위로 자유롭게 선정할 수 있으나, 본 과제에서는 입력 전압을 DC 24V를 사용하였다. 본 센서의 검출 범위는 0 ~ 10,000ppm이고, 허용 오차는 ±1ppm이다. 그림4-8은 에틸렌 센서의 외형이다.

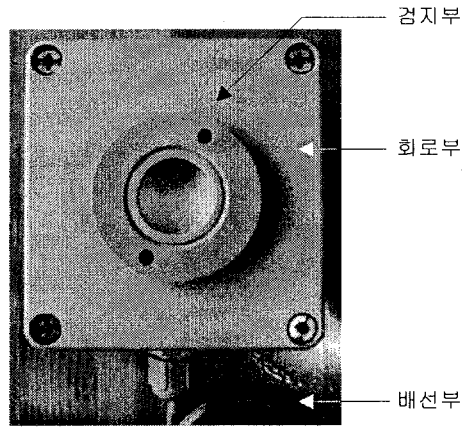


그림4-8. O<sub>2</sub>(산소) 센서

#### 5. 센서 연결

본 과제에서 사용된 센서들의 연결 법은 4 선식 방법과 2 선식 방법으로 구분되며, 각각 방식에서 사용되는 방법으로는 전류 모드와 전압 모드 두 가지 방식으로 나뉘어 있다.

##### 가. 2 선식 방법의 동작 방법

###### 1) 전류 모드

전류모드는 그림4-9과 같이 구성이 된다. 발신기의 흑색 단자(루프 -)와 전원장치의 (-)측을 외부 기기에 연결하여 전류 루프를 구성한다. 이때, 루프를 통하여 흐르는 전류는 식4-1과 같이 되고 측정된 전류로부터 해당 가스의 농도를 환산하려면 식4-2를 사용한다.



$$I(\text{mA}) = \frac{16 \times \text{가스농도}[\text{PPM}, \%]}{\text{측정범위}[\text{PPP}, \%]} + 4 \quad (\text{식 4-1})$$

$$\text{가스농도}[\text{PPP}, \%] = \frac{(I[\text{mA}] - 4) \times \text{측정범위}[\text{PPPM}, \%]}{16} \quad (\text{식 4-2})$$

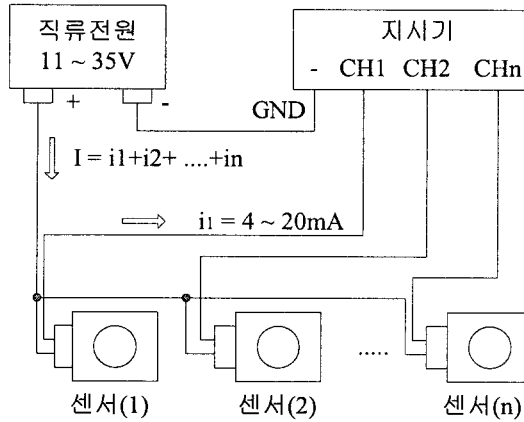


그림4-9. 2 선식 전류 모드 결선도

## 2) 전압 모드

전압 모드로 사용할 경우에는 적절한 용량의 부하저항( $R_L$ )를 이용하여 그림4-9와 같이 전류 Loop를 구성한다. Loop를 통하여 흐르는 전류는 위에 식4-1과 같고 부하저항 양단에 나타나는 전압은 식4-3로 나타낼 수 있다. 그리고 해당 농도를 환산하려면 식4-3를 사용하여 계산한다. 그림4-10은 전압 모드 결선도를 나타내었다.

$$V[\text{mV}] = \left\{ \frac{16 \times \text{가스농도}[\text{PPM}, \%]}{\text{측정범위}[\text{PPM}, \%]} + 4 \right\} \times R_L[\Omega] \quad (\text{식 4-3})$$

$$\text{가스농도}[\text{PPM}, \%] = \frac{(V[\text{mV}] - 4 R_L) \times \text{측정범위}[\text{PPM}, \%]}{16 R_L} \quad (\text{식 4-4})$$

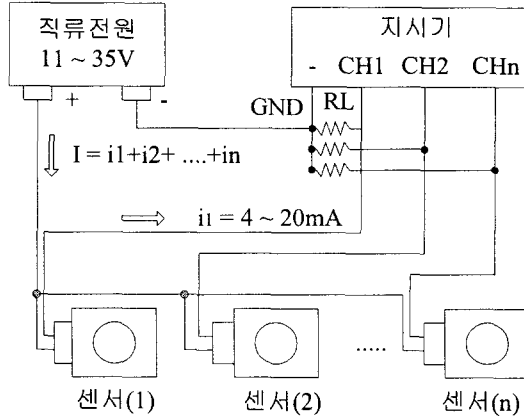


그림4-10. 2 선식 전압 모드 결선도

나. 4 선식 방법의 동작 방법

본 과제에 사용된 C<sub>2</sub>H<sub>4</sub>(에틸렌) 센서는 4선 전송 방식의 연결도로 구성된다. 4선 전송 방식은 전류 모드와 전압 모드 두 방식으로 구분되며, 본 과제에서는 전압 모드를 사용하였다.

1) 전류 모드

전류 모드로 사용할 경우에는 그림4-11과 같이 발신기의 백색 단자(루프 +)와 회색 단자(루프 -)를 이용하여 전류 루프를 구성한다. 이때 가스 농도의 측정 범위는 0 ~ 100 LEL이므로 루프를 통하여 흐르는 전류는 식4-5와 같으며, 측정된 전류로부터 가스의 농도를 환산하려면 식4-6을 사용한다.

$$I_{[mA]} = \frac{16 \times \text{가스농도}[\%LEL]}{100} + 4 \quad (\text{식 4-5})$$

$$\text{가스농도}[PPM, \%] = 6.25 \times I_{[mA]} - 25 \quad (\text{식 4-6})$$

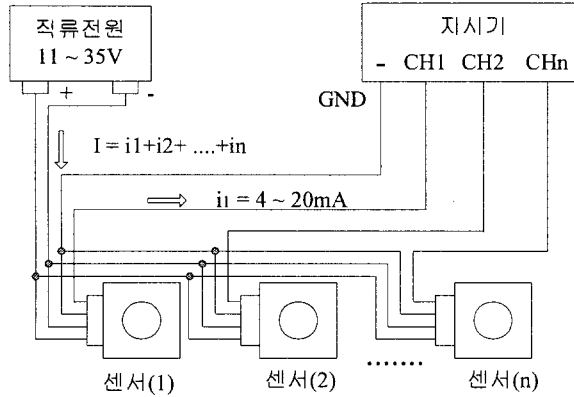


그림4-11. 4 선식 전류 모드 결선도

2) 전압 모드

전압 모드로 사용할 경우, 적절한 용량의 부하저항을 이용한다. 그림4-12에 흐르는 전류는 식4-5와 같고, 부하 저항 양단에 걸리는 전압은 식4-7과 같이 계산된다. 부하저항 양단의 전압을 측정 후, 식4-8를 이용하여 가스 농도를 구한다.

$$V[\text{mV}] = \left\{ \frac{16 \times \text{측정농도}[\%LEL]}{100} + 4 \right\} \times R_L[\Omega] \quad (\text{식 4-7})$$

$$\text{측정농도}[\%LEL] = \frac{6.25 \times V[\text{mV}]}{R_L[\Omega]} - 25 \quad (\text{식 4-8})$$

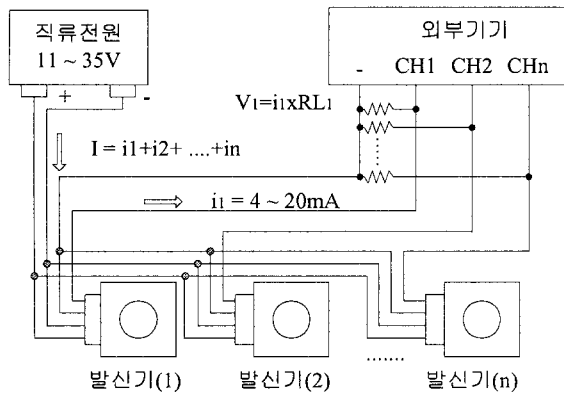


그림4-12. 4 선식 전압 모드 결선도

### 제3절 센서 제어기의 구성

센서 제어기의 구성은 측정된 센서값이 입력되는 센서값 입력부, 아날로그 신호를 입력받아 디지털로 변환하는 ADC부, 센서 제어기 시스템 동작을 위한 프로그램이 저장되어 있는 ROM, 시스템에 전원을 공급하는 전원부, 전체 시스템의 제어를 담당하는 제어부, 중계기와 데이터 송수신을 담당하는 데이터 송수신부로 이루어져 있다. 그림4-13은 전체 블록도를 나타낸 것이다.

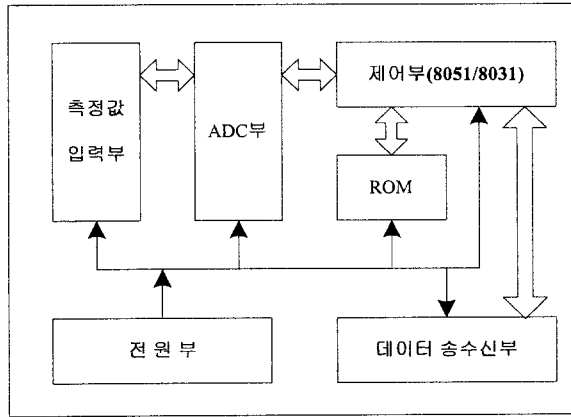


그림4-13. 센서 제어부 블록도

#### 1. 센서모듈 입력부

센서모듈 입력부는 센서에서 측정된 전류 데이터 값을 전압 데이터로 변환하는 부분이다. 필터부는 RCV420을 통하여 출력된 전압의 노이즈 성분을 제거하여 정확한 데이터를 저장하는 역할을 수행한다. 변환된 데이터는 ADC(Analog-to-Digital Converter)의 입력부의 입력으로 사용된다. 센서 모듈부에 사용되는 소자들의 기능을 간략하게 다음에 간략하게 설명하였다.

##### 가. RCV420

본 과제에 사용된 RCV420은 입력으로 들어오는 전류를 전압으로 변환하는 IC이다. 전류 4 ~ 20mA를 입력으로 받아 전류 값에 비례하여 출력 전압 0 ~ 5V를 출력하며 허용오차는 0.1%이다.

## 나. LM741

LM741은 Low Pass Filter를 구성하기 위한 능동소자로서, RCV420을 통하여 출력된 전압의 노이즈 성분을 제거하여 정확한 데이터를 저장하는 역할을 수행한다.

## 2. ADC부

ADC부는 센서 모듈부의 입력부의 입력으로 들어오는 아날로그 데이터 신호를 디지털 신호로 변환하는 역할을 수행한다. 본 과제에서는 ADC0809를 사용했다. ADC0809는 8bit 아날로그 신호를 입력받아 8bit 디지털 신호를 출력한다. 본 IC는 8-채널 멀티플렉서와 마이크로 콘트롤러 로직이 하나의 IC로 통합되어 있다. 입력 신호에 대한 선택은  $8 \times 1$  멀티플렉스에 연결된 3비트 어드레스에 선택된다.

## 3. 제어부

제어부는 센서제어보드의 전체적인 제어를 담당하는 역할을 수행한다. 이 부분은 마이크로 콘트롤러인 8051/8031과 메모리 부분으로 구성되어 있다. 먼저 8051/8031에 프로그램 메모리 영역에 직렬 통신에 초기값을 정해주고 입력 받고자하는 센서부를 선택후, 센서의 데이터 값을 중계기에 전송하는 역할을 수행한다. 74LS573는 데이터의 보호를 위하여 사용되었고, 74LS73는 ADC0809에 클럭신호를 보내 준다.

### <8051/8031>

마이크로 프로세서로 8051/8031을 사용하였다. 8051은 제어응용에 적합한 8비트 CPU로서 32개의 양방향성입출력 핀을 가지고 있다. 128비트의 데이터 RAM과 4K비트의 내부 ROM이 내장되어 있다. 64KB의 외부 프로그램 메모리 어드레스 영역과 데이터 메모리 어드레스 영역이 각각 존재한다. 8031은 자체 ROM이 없고 모든 명령을 외부 메모리로부터 제공받는다라는 점을 제외하고는 8051과 모든 기능이 같다.

## 4. ROM

ROM은 센서제어보드의 동작에 필요한 프로그램이 내장되어 있는 부분이다. 본 과제에 사용된 프로그램은 어셈블러를 사용하였으며, 29C256 IC를 사용하였다. 29C256은 CMOS Flash Memory로서 256K( $32K \times 8$ )이다. 15개의 어드레스핀과 8개의 Data I/O 핀으로 이루어져 있다.

## 5. 전원부

전원부는 센서제어보드 전체에 전원을 공급하는 부분이다. 입력 전압은 24V와 12V 두 종류에 전원을 사용한다. 24V는 센서 전원과 5V 전원을 얻기 위한 전원이고, 12V는 -15V전원을 얻기 위한 전원으로 사용되었다. 센서에서 온/습도 센서는 -15V를 입력 전원 사용하고 그 외의 모든 센서는 24V를 입력 전원으로 사용하였다. 전원부의 구성은 DC/DC 컨버터, 코일, LM2575 등으로 이루어져 있다.

### 가. DC/DC 컨버터

DC/DC 컨버터는 입력으로 들어오는 전원을 사용자가 원하는 전원으로 변환해주는 역할을 한다. 본 과제에 쓰인 DC/DC 컨버터는 입력 전압 +12V를 입력받아  $\pm 15V$ 전원으로 변환을 시켜준다.

### 나. LM2575

LM2575는 정전압 소자로써 입력되는 전압을 시스템에 맞는 전압으로 변환을 시켜준다. 입력 전원을 7 ~ 40V까지 입력받아 출력 전압으로 +5V를 출력한다. 본 과제에서는 24V를 입력받아 출력 전압 +5V를 출력하도록 선택하였다.

## 6. 데이터 송수신부

데이터 송수신부에서는 제어부에서 전송된 데이터를 직렬 통신의 규격에 맞게 변환을 시켜준다. 본 과제에서는 이 규격을 맞는 데이터를 얻기 위해서 MAX232를 사용하였다. 본 과제에서는 TTL(0 ~ 5V) 레벨의 데이터를 -12V~+12V의 데이터로 변환시켜서 중계보드에 데이터를 전송한다. 그림4-14에 센서 제어부 보드를 나타내었다.

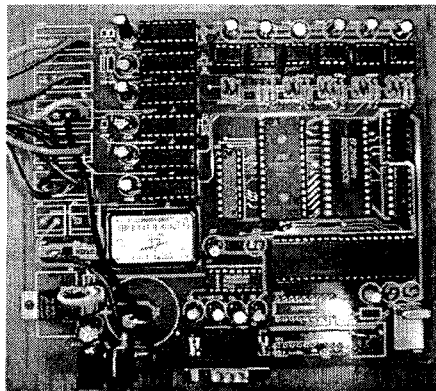


그림4-14. 센서 제어부 보드

<순 서 도>

센서 제어기는 전원이 인가되면 초기에 중계기로 데이터 전송을 위하여 직렬 통신을 동기화 한다. 센서 제어기는 중계기로부터 센서 모듈 선택 신호를 수신 후, 선택된 센서 데이터를 측정한다. 측정된 데이터를 중계기에 전송하기 위하여 전류를 전압으로 변환후 중계기로 데이터를 전송한다. 그림4-15에 전체적인 센서 제어기의 순서도를 나타내었다.

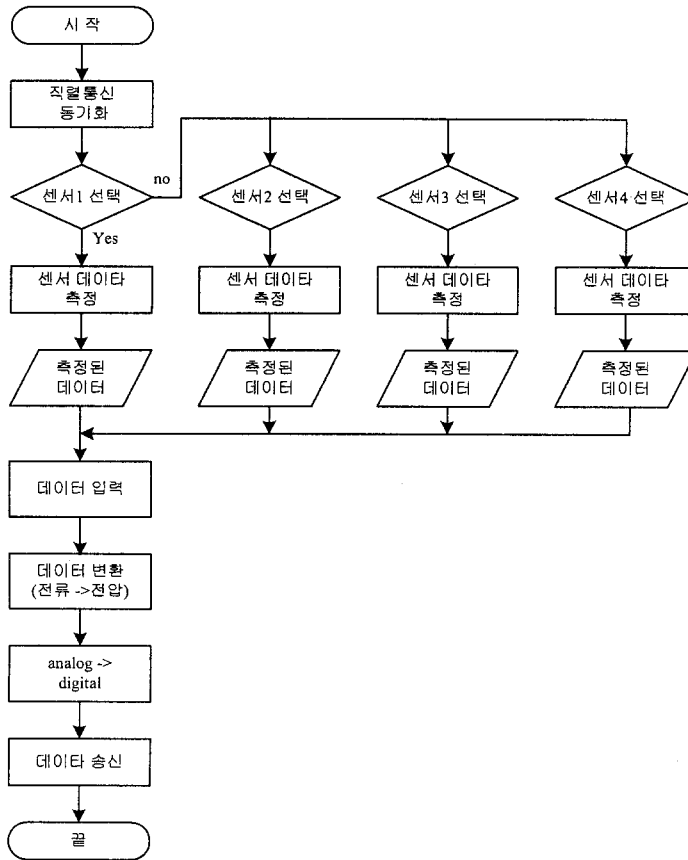


그림4-15. 센서 제어기의 순서도

## 제4절 중계기의 구성

중계기의 구성은 아날로그 믹스(Analog Mux)부, 제어부, 메모리부, HOST PC와 데이터 송수신을 위한 데이터 송수신부 및 전원부로 구성된다. 아날로그 믹스는 여러대의 센서 모듈중 특정한 센서 모듈을 선택하여 제어부로 데이터를 전송하고, 중계기 시스템 동작을 위하여 프로그램 저장된 롬, 전체적인 시스템을 제어하기 위한 제어부와 전원 공급을 위한 전원부로 구성되어 있다. 중계기와 HOST PC사이의 데이터 송수신을 담당하는 데이터 송수신부로 구성되어 있다. 그림4-16은 전체적인 중계기의 구조를 도시하였다. 향후, 센서 제어기 증가에 대비하여 중계기의 입력단에 대용량의 아날로그 믹스를 사용하였다. 중계기는 HOST PC로부터 특정한 센서 모듈의 입력선택 명령을 수신 후, 센서 모듈을 선택한다. 제어부는 센서 모듈로부터 측정된 센서 데이터를 저장하는 역할을 수행한다. 저장된 데이터는 RS-232 인터페이스를 통하여 HOST PC로 송신하게 된다. 본 과제에서는 안정된 전원회로를 설계 및 제작하였다. 기존의 전압 안정화 회로에서 사용하는 LM7805를 LM2575로 대체하고 코일을 사용하였다.

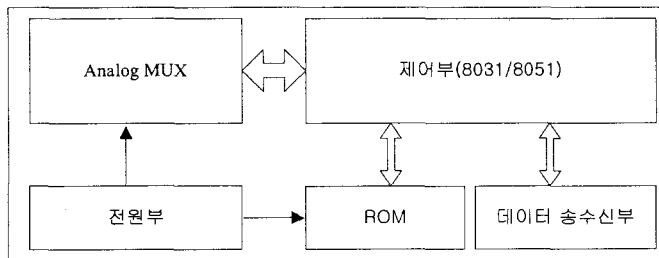


그림4-16. 센서 중계기의 구성도

### 1. 제어부

제어부는 수신된 데이터와 송신할 데이터의 처리를 위해 8051 계열의 마이크로 프로세서를 사용하였다. 이는 가장 범용으로 사용되는 제어용 칩으로 직렬 통신 및 데이터 입출력을 통한 데이터 처리기능을 포함하고 있다.

### 2. 데이터 송수신부

데이터 송수신부에서는 제어부로 부터 전송된 직렬 데이터를 직렬 통신 규격으로 변환



시키는 기능을 수행해야만 한다. 이를 위해 MAXIM 사의 MAX232 칩을 사용하였다. 이는 TTL (0 ~ 5V) 레벨의 데이터를 -12V~+12V의 데이터로 변환시켜서 HOST PC로 전송시켜 준다.

### 3. ROM

ROM은 실제 증계기 제어를 위한 프로그램을 저장하는 장치이다. 이를 위해 ATMEL사의 29C256을 사용하였으며, 사용된 ROM은 EEPROM 형태로 프로그램의 수정이 용이하여 향후 업그레이드가 가능하다.

### 4. 아날로그 멀티플렉서부

아날로그 멀티플렉서부는 데이터 송수신부에서 수신된 각 센서 모듈의 직렬 데이터를 제어부에 의해 선택해서 선택된 직렬 데이터를 다시 HOST PC로 전송하는 역할을 수행한다. 이를 위해 Analog Device 사의 ADG408을 사용하였다.

### 5. 전원부

본 과제에서는 안정된 전원회로를 설계 및 제작하였다. 기존의 전압 안정화 회로에서 사용하는 LM7805를 LM2575로 대체하고 코일을 사용하였다. 그림4-17에 제작된 증계보드를 나타내었다.

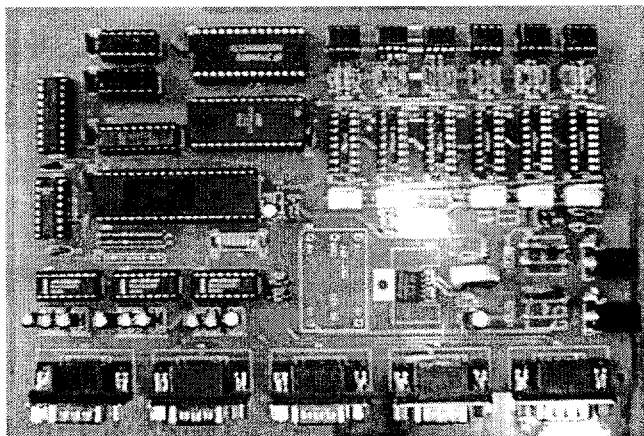


그림4-17. 센서 증계기 보드

## 6. 순서도

중계기에 전원이 인가되면 초기에 센서 제어기에 센서 모듈 선택을 위하여 직렬 통신을 초기화한다. 중계기는 센서 모듈 선택 신호를 송신후, 측정된 데이터를 센서 모듈부로부터 수신한다. 센서 모듈부로부터 수신된 데이터를 중계기를 통하여 호스트 PC로 데이터를 송신한다. 그림4-18에 전체적인 센서 제어기의 순서도를 나타내었다

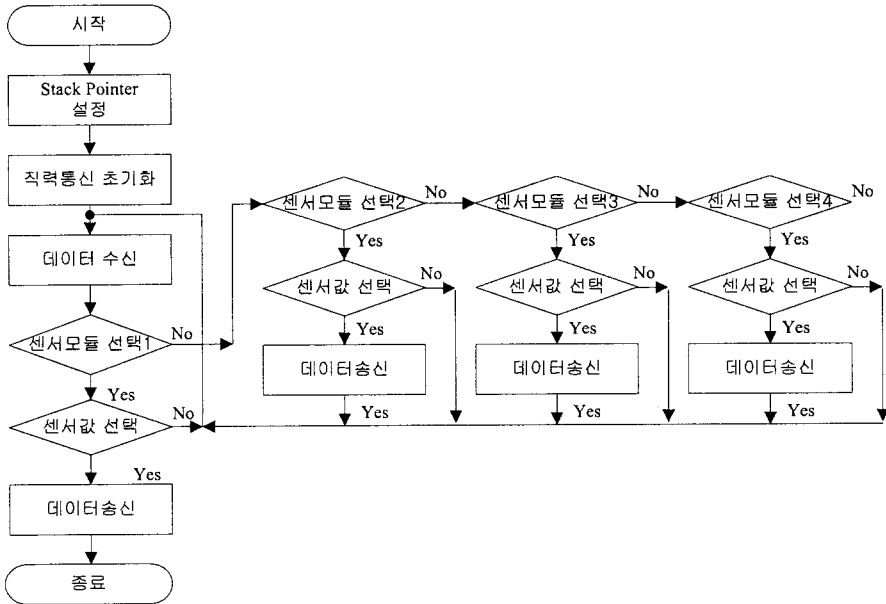


그림4-18. 중계기용 프로그램의 순서도

## 제5절 통신 프로토콜의 구성

그림4-18은 HOST PC와 중계기간의 통신 프로토콜을 도시하였다. 그림4-19(a)는 HOST PC로부터 중계기로 송신되는 신호로 데이터의 길이는 두 바이트로써 상위 한 바이트는 저장고의 주소를 나타내고 하위 한 바이트는 센서 모듈부의 주소를 나타낸다. 그림4-19(b)는 중계기로부터 HOST PC로 송신되는 신호로 데이터의 길이는 한 바이트로써 측정된 센서 데이터 값을 송신하게 된다.

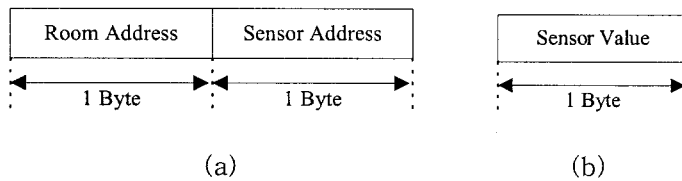


그림4-19. HOST PC 및 중계기간의 데이터 신호 구조

HOST PC는 저장고 및 센서 모듈신호를 중계기로 전송된다. 중계기는 HOST PC가 요구하는 저장고 및 센서 모듈을 선택한다. 선택된 센서 제어기는 센서 모듈로부터 데이터를 측정하여 측정된 데이터를 중계기로 송신한다. 중계기는 센서 제어기로부터 송신된 측정된 데이터를 HOST PC로 전송하게 된다. HOST PC는 중계기에서 송신된 측정 데이터를 모니터 상에 디스플레이 하는 기능을 수행한다. 그림4-20는 전체 시스템의 흐름도를 도시하였다. 본 과제의 전체 시스템 응용 예를 그림4-21에 나타내었고 센서부, 중계기, HOST PC로 구성되어 진다. 센서제어기와 중계기, 중계기와 HOST PC 간의 통신 프로토콜은 마스터-슬레이브 구조로 설계하였다. 센서부 및 중계기의 제어부는 인텔의 인텔의 8051 8비트 마이크로 컨트롤러를 채용하였다. 마이크로 프로세서를 채용함으로써 샘플 간격을 계절이나 시간대별로 구분하여 적용 가능하며, 시스템의 안정성을 향상시키고, 센서제어기와 중계기를 모듈화 함으로써 유지보수 및 시스템 향상을 가능하도록 설계 및 제작되었다.

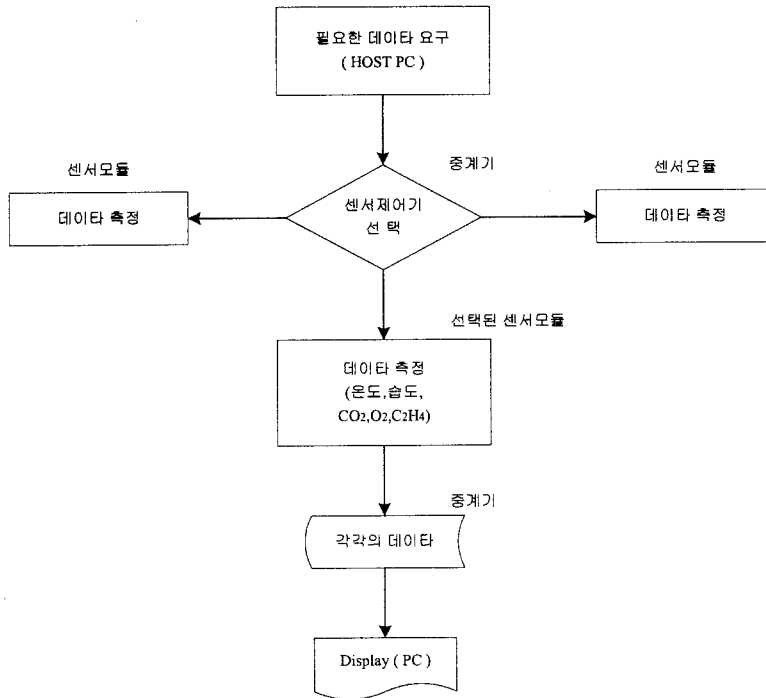


그림4-20. 시스템 동작 순서도

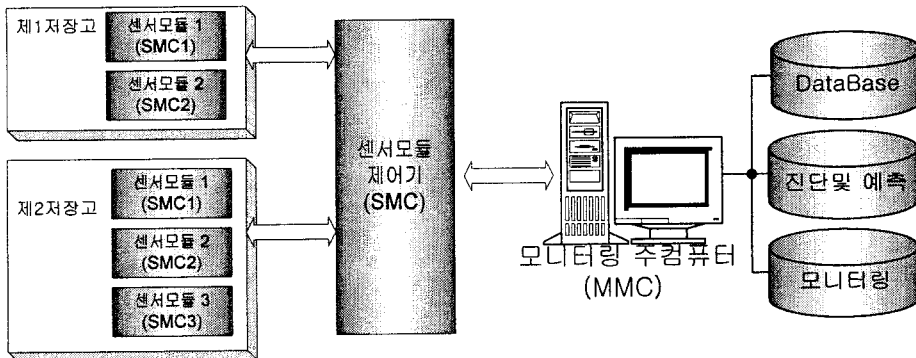


그림4-21. 저온저장 자동화 시스템의 구성 예

## 제 6 절 결론

현재 국내에 가장 많이 보급 되어있는 저장방식은 저온저장고이며, 현재 계속적으로 증가 추세에 있는 실정이다. 그러나 저장고내의 환경을 자동적으로 관리하고 조절할 수 있는 전산화 및 관련 하드웨어 구축은 전무한 상태이다. 현재 저온저장고에는 온도와 습도 센서는 설치되어 사용 있지만 산소와 이산화탄소 센서는 설치 되어있지 않다. 이러한 문제점으로 인하여 저온저장고에 따른 환경조성과 저장중 저장요인의 변화를 수시로 관리하지 못하므로 저온저장고의 면적 증가 추세에도 불구하고 피해가 해마다 증가하는 실정이다.

이러한 문제점을 개선하기 위하여 저온 저장고내 환경, 즉 온도, 습도, 이산화탄소, 산소 그리고 에틸렌에 대한 농도 변화에 대한 정보를 각 저장고 별로 검사하여 호스트 컴퓨터로 전송한 뒤 자료를 축적함으로써 과수의 관리 및 상품성을 예측하여 저장 손실을 최소화하여 국가경쟁력 향상 및 부가 가치 창출을 할 수 있다.

1 차년도에서는 센서 자료 수집 및 분석하였고, 이를 통하여 가스 측정 시스템의 하드웨어 제작 및 검증을 수행하였다. 2 차 년도에서는 1 차년도의 연구 결과를 바탕으로 하여 센서 동작 표시 기능과 센서 고장 점점 및 고장 경고 신호 기능을 내장한 센서 제어기 설계 및 검증을 하였다. 3 차년도에서는 센서 모듈부의 안정화와 측정된 데이터의 오차 최소화를 위하여 센서보드 및 중계보드의 안정성을 고려하여 향상된 시스템을 설계 및 제작하여 측정 데이터의 오차 최소화 및 시스템 안정화 검증을 수행하였다.

## 참 고 문 헌

ANDRICH, G., R. FIORENTINI, A. TUCI und C. GALOPPINI, 1989. Skin permeability to oxygen in apples stored in controlled atmosphere. J. Amer. Soc. Hort. Sci. 114, Vol. 5, 770-775.

ARGRÜNER, B. 1992. Energie und Lagerungstechnische Untersuchungen Zur CA-Lagerung von A'pfeln. Vesseres Obst. 2: 4-7.

BANKS, N.H. und S.J. KAYS, 1988 . Measuring internal gases and lenticel resistance to gas diffusion in potato tubers. J. Amer. Soc. Hort. Sci. 113, Vol. 4, 557-580.

BAUMANN, H.,1990. Determination of Gas Exchanges in 'CA' using closed chambers. Air pollution research report, 26 : 93-96.

BEN YEHOSHUA, S., R.N. ROBERTSON und J.B. BIALE, 1993 . Respiration and internal atmosphere of the avocado fruit. Plant Physiol. 38, 194-201.

BIALE, J.B., 1960 . Handbuch der Pflanzenphysiologie. 12, Vol. 2, 536-592.

BOEKE, J. E., 1968. Evolution of post-harvest trends of respiration rates and softening of apples and tomato. Mededeliugen van de Landbouwhogesschool, Wageningen 68-15, 1-89.

BOHLING, H. und H. HANSEN, 1974 . Respiration measurements in various kinds of vegetables and fruit during storage under increased CO<sub>2</sub> and reduced O<sub>2</sub> concentrations. Acta. Hort. 38, Vol. 1, 83-87.

BRÄNDLE, R.,1968 . Die Verteilung der Sauerstoffkonzentrationen in fleischigen Speicherorganen (Äpfel, Bananen and Kartoffelknollen). Berichte der Schweizerischen Botanischen Gesellschaft. 78, 330-365.

- BURG, S.P. und E.A. BURG, 1962. Role of ethylene in in fruit ripening. *Plant Physiol.* 37, 179-189.
- BURG, S.P. und E.A. BURG, 1965 . Gas exchange in fruits. *Physiologia Plantarum.* 18, 870-884.
- BURNS, M., 1978 . New D25 textile shade monitor. Hunterlab reflections and transmissions, Hunter associates laboratory, Inc. 26, 1-2.
- BURTON, W.G. und W.T. SPRAGG, 1950 . A note on the intercellular space of the potato tuber. *New Phytologist.* 49, 8-10.
- BURTON, W.G., 1974 . Some biophysical principles underlying the controlled atmosphere storage of plant material. *Ann. Appl. Biol.* 78, 149-168.
- BURTON, W.G., 1982 . Postharvest physiology of food crops. London und New York.
- BUSSEL, J. und E.C. MAXIE, 1966 . Gas exchange in Bartlett pears in relation to gamma irradiation . *Amer. Soc. Hort. Sci.* 88, 151-159.
- CAMERON A.C. und S.F. YANG, 1982 . A simple method for the determination of resistance to gas diffusion in plant organs. *Plant Physiol.* 70, 21-23.
- CAMERON, A.C. und M.S. REID, 1982 . Diffusive Resistance : Importance and measurement in controlled atmosphere storage. In : D.G. Richardson, M. Meheriuk (ed.). *Controlled atmospheres for storage and transport of perishable agricultural commodities.* 171-178.
- COREY, K.A., A. EYDELAND und S.A. ALTMANN, 1989 . Mathematical model for predicting oxygen concentration gradients in muskmelon fruits. *Acta Hort.* 258, 199-206.
- COREY, K.A., D.M. PHARR und H.P. FLEMING, 1983 . Role of gas diffusion in bloater formation of brined cucumbers. *J. Food Sci.* 48, 389-399.

CZICHOS, H. (ed.), 1989 . Hütte : die Grundlagen der Ingenieurwissenschaften. Akademischer Verein Hütte e.V. Berlin.

DADZIE, B.K., N.H. BANKS, E.W. HEWETT und D.J. CLELAND, 1990 . Variation in internal atmosphere composition within single apples. Abstracts contrib. papers. 23rd Int. Hort. Congr., Firenze. 1, 659.

DEVAUX, M.H., 1981 . Étude expérimentale sur l'aération des tissus massifs. Annl. Sci. nat. (Bot.) 7 ser. 14, 297-395.

FIDLER, J.C. und C.J. NORTH, 1971 . The effect of conditions of storage on the respiration of apples. 5. The relationship between temperature, rate of respiration and composition of the internal atmosphere of the fruit. J. Hort. Sci., 46, 229-235.

FORSYTH, F.R., I.V. HALL und H.J. LIGHTFOOT, 1973 . Diffusion of CO<sub>2</sub>, O<sub>2</sub> and C<sub>2</sub>H<sub>4</sub> in cranberry fruit. HortScience. 8, 45-46.

GARTE, L. und J. WEICHMANN, 1974 . Storage ability of pickling cucumbers as influenced by the method of harvesting. Acta. Hort. 32, Vol. 2, 373-377.

HENZE, J., 1983. Criteria to determine the quality of apples dependent on harvesting date and storage treatments. Acta Horticulture 138, 99. 69-75.

HENZE, J. und C. HURTER, 1980 . Respiration and internal atmosphere in "vegetable fruits" as influenced by storage conditions. Acta. Hort. 116, 41-46.

HENZE, J. und H. HANSEN, 1988 . Lagerräume für Obst und Gemüse. KTBL-Schrift 327.

ISENBERG, F.M.R., 1979 . Controlled atmosphere storage of vegetables. Hort. Rev., Vol. 1, 373-394.

KADER, A.A., 1985 . An overview of the physiological and biochemical basis of CA



effects on fresh horticultural crops. In : S.M. Blankenship (ed.). Proc. 4th Natl. CA Res. Conf. North Carolina State Univ., Raleigh. Hort. Rep. 126, 1-9.

KADER, A.A., 1989 . Respiration and gas exchange of vegetables. In : J. Weichmann (ed.). Postharvest physiology of vegetables. 25-43.

KIDD, F. und C. WEST, 1935a . Resistance of the skin of the apple fruit to gaseous exchange. Rep. Food Invest. Board 1934. 59-64.

KIDD, F. und C. WEST, 1935b . Carbon dioxide injury in relation to the maturity of apples and pears. Rep. Food Invest. Board 1934. 64-68.

LALAGUUA, F. AND S. THORNE ; gas exchange between atmosphere of different oxygen concentration and "Cox's orange pippin" apples. Oregon State University. Symposium of "CA for storage and transport of perishable Agricultural commodities". Timber press. pp. 155-160.

MARCELLIN, P., P. SOUDAIN und PHAN PHUC, 1967 . Le gas carbonique des tissus des fruits étudié par perfusion d'air ou d'anhydride carbonique. Bull. Soc. Franc. Physiol. Végét. 13, 33-41.

MOHR, H. und P. SCHOPFER, 1985. Lehrbuch der Pflanzenphysiologie. Berlin, Heidelberg, New York, Tokyo.

박윤문, 1996. : 'Fuji' 사과와 CA 저장중 발생하는 내부 갈변 현상. 원예산물의 저장과 유통. 6. 22-24.

RAJAPAKSE, N.C., N.H. BANKS, E.W. HEWETT und D.J. CLELAND, 1990. Development of oxygen concentration gradients in flesh tissues of bulky plant organs. J. Amer. Soc. Hort. Sci. 115, Vol. 5, 793-797.

RYALL, A.L. und W.J. LIPTON, 1979. Handling, transportation and storage of fruits and vegetables. 1. Vegetables and melons. AVI publishing Co. Inc., Westport.

SALTVEIT, M.E., 1985. A summary of CA requirements and recommendations for vegetables. In : S.M. Blankenship (ed.). Proc. 4th Natl. CA Res. Conf. North Carolina State Univ., Raleigh. Hort. Rep. 126, 471-492.

SASTRY, S.K. und D.E. BUFFINGTON, 1983 . Transpiration rates of stored perishable commodities : A mathematical model and experiments on tomatoes. Int. J. Refrig. 6, 84-96.

SMITH. W.H., 1947 . A new method for the determination of the composition of the internal atmosphere of fleshy plant organs. Ann. bot. 11, Nr. 49, 363-368.

SOLOMOS, T., 1982 . Effects of low oxygen concentration on fruit respiration : Nature of respiratory diminution. In : D.G. Richardson, M. Meheriuk (ed.). Controlled atmospheres for storage and transport of perishable agricultural commodities. 161-170.

SOLOMOS, T., 1987 . Principles of gas exchange in bulky plant tissues. HortScience. 22, Vol. 5, 766-771.

SOLOMOS, T., 1989. A simple method for determining the diffusivity of ethylene in 'McIntosh' apples. Scientia Hort. 39, 311-318.

STILES W., 1960. The composition of the atmosphere (oxygen content of air, water, soil, intercellular spaces, diffusion, carbon dioxide and oxygen tension). Handbuch der Pflanzenphysiologie. Berlin, Göttingen und Heidelberg. 12, vol. 2, 114-148.

STRASBURGER, E. (ed.), 1983. Lehrbuch der Botanik. Stuttgart und New York.

WANNER, H., 1945. Studies in tropical als begrenzender Faktor der Atmung von Pflanzenwurzeln. Vierteljahresschr. Naturf. Ges. 90, 98-119.

WEICHMANN, J., 1971. Automatische Anlage für Gasstoffwechselfmessungen von Gemüse. Gartenbauwissenschaft. 36, 251-258.

WEICHMANN, J., 1972. Lagerung und Gasstoffwechsel bon Möhren (Daucus carota L.) in veränderter Atmosphäre. Dissertation, TU-München/Weihenstephan.

WEICHMANN, J., 1977. Physiological response of root crops to controlled atmospheres. Michigan State Hort. Rep. 28, 122-136.

WOLLIN, A.S., C.R. LITTLE und J.S. PACKER, 1985. Dynamic control of storage atmospheres. In : S.M. Blankenship (ed.). Proc. 4th Natl. CA Res. Conf. North Carolina State Univ., Raleigh. Hort. Rep. 126, 308-315

D.A. BELL, Electronic Instrumentation and Measurements 2nd Ed., Prentice Hall, 1994

J.M. JACOB, Industrial Control Electronic Devices and Circuit Theory, Prentice Hall, 1996

R. BOYLESTAD, L. NASHELSKY, Electronic Devices and Circuit Theory, Prentice Hall, 1996

T.L. FLOYD, Electronic Devices, 4th Ed., Prentice Hall, 1996

R.J. TOCCI, Digital Systems - Principles and Applications, 6th Ed., Prentice Hall, 1995

Microsoft Visual C++ 6.0 Reference Library(BK+CD), Microsoft-press 1998

유희정, 박병수, 최병렬, 양용준 “저장 원예 작물의 환경관리 자동화 시스템 개발”, 한국 정보처리 학회, '99 추계학술대회 논문집, p.104~p.108, 1999

이상엽, Visual C++ Programming Bible Ver 6.x, 영진출판사, 1998

Java2 Programming Bible, 정보문화사, 손진욱 편저, 1999

김경만, ASP로 웹 서버 구축하기, 비비컴, 1997

클릭하세요 ASP 프로그래밍, 권영만

<http://www.msdn.microsoft.com/default.asp>

<http://www.jacasoft.com>

D.A BELL, Electronic Instrumentation and Measurements 2nd Ed., Prentice Hall, 1994

J.M JACOB, Industrial Control Electronic Devices and Circuit Theory, Prentice Hall ,1996

R. BOYLESTAD, L. NASHELSKY, Electronic Devices and Circuit Theory, Prentice Hall ,1996

T.L FLOYD, Electronic Devices, 4th Ed., Prentice Hall ,1996

R.J TOCCI, Digital Systems - Principles and Application, 6th Ed., Prentice Hall ,1995

W. GOPEL, J. HESSE, J.N. ZEMEL, Sensors a Comprehensive Survey, VCH

Schultz, Thomas W, C and the 8051, Prentice Hall, 1999

MacKenzie, I. Scott, The 8051 Microcontroller , Prentice Hall, 1998

8051 C Compiler Programming Guide

Predko, Myke, Programming and Customizing the 8051 Microcontroller, McGraw-Hill, 1999

Stewart, James W./ Miao, Kai/ Miao, Kai X., The 8051 Microcontroller, Prentice Hall, 1998

MacKenzie, I. Scott, The 8051 Microcontroller, Prentice Hall, 1998년

Adams, Martin David, Sensor Modelling, Design and Data Processing for Autonomous Navigation, World Scientific Pub Co, 1998

Ristic, Ljubisa (Edt), Sensor Technology and Devices, Artech House, 1994

최병재, 양승호, 마이크로컨트롤러8051, 홍릉과학출판사, 1998

김현기외2인, 8051프로그래밍, 생능출판사

이동렬, 김중태, 8051길라잡이-기초와응용, 두남문화사, 1997

문세홍, 나종래, 인텔8051구조와설계응용, OHM사. 1989

제어용 마이컴 초보에서 응용까지, 성안당, 1992

박선호, FA 센서 응용백과(1), 영진출판사, 1993

박선호, FA 센서 응용백과(2), 영진출판사, 1993

차홍식, 센서이론과실험, 일진사, 1998

## 부 록

### 1. 환경관리를 위한 자동측정 응용 소프트웨어 개발 프로그램

```
// nong2_11View.cpp : implementation of the CNong2_11View class
#include "stdafx.h"
#include "nong2_11.h"
#include "nong2_11Set.h"
#include "nong2_11Doc.h"
#include "nong2_11View.h"
#include "option.h"
#include <time.h>
#include "data.h"
#include "stropt.h"

#define TEMP (60.000000/255.000000)*float(in)-5
#define HUM (100.000000/(255.000000))*float(in)
#define CARBON (25.000000/255.000000)*float(in)
#define OXY (25.000000/255.000000)*float(in)
#define ETIL (10000.000000/255.000000)*float(in)

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CNong2_11View

IMPLEMENT_DYNCREATE(CNong2_11View, CDaoRecordView)

BEGIN_MESSAGE_MAP(CNong2_11View, CDaoRecordView)
//{{AFX_MSG_MAP(CNong2_11View)
ON_MESSAGE(WM_COMM_RXCHAR, OnCommunication)
ON_BN_CLICKED(IDC_BUTTON1, OnData)
ON_BN_CLICKED(IDC_START, OnStart1)
ON_BN_CLICKED(IDC_BUTTON6, OnButton6)
}}
```

```

ON_BN_CLICKED(IDC_BUTTON3, OnStrOpt)
ON_BN_CLICKED(IDC_FILEPRINT, OnFileprint)
ON_WM_TIMER()
ON_BN_CLICKED(IDC_BUTTON4, OnInfo)
ON_BN_CLICKED(IDC_BUTTON2, Ongraph)
ON_BN_CLICKED(IDC_ADDSA, OnAddsa)
ON_BN_CLICKED(IDC_DATASHOW, OnDatashow)
//}}AFX_MSG_MAP
// Standard printing commands
ON_COMMAND(ID_FILE_PRINT, CDaoRecordView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_DIRECT, CDaoRecordView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_PREVIEW, CDaoRecordView::OnFilePrintPreview)
END_MESSAGE_MAP()

```

```

////////////////////////////////////
// CNong2_11View construction/destruction

```

```

CNong2_11View::CNong2_11View()
: CDaoRecordView(CNong2_11View::IDD)

```

```

{
//{{AFX_DATA_INIT(CNong2_11View)
m_pSet = NULL;
m_ptime1 = _T("");
m_ptemp1 = 0;
m_poxy1 = 0;
m_pcarbon1 = 0;
m_petil1 = 0;
m_phum1 = 0;

m_ptime2 = _T("");
m_ptemp2 = 0;
m_poxy2 = 0;
m_pcarbon2 = 0;
m_petil2 = 0;
m_phum2 = 0;

```

.....

```

        m_dsnum = 0;
        m_dssee = 0;
        m_dsdate = (DATE)0;

        //}}AFX_DATA_INIT
        // TODO: add construction code here
    }

CNong2_11View::~CNong2_11View()
{
}

void CNong2_11View::DoDataExchange(CDataExchange* pDX)
{
    CDaoRecordView::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CNong2_11View)
    DDX_Control(pDX, IDC_BUTTON6, m_btn6);
    DDX_Control(pDX, IDC_START, m_btn5);
    DDX_Control(pDX, IDC_BUTTON4, m_btn4);
    DDX_Control(pDX, IDC_BUTTON3, m_btn3);
    DDX_Control(pDX, IDC_BUTTON2, m_btn2);
    DDX_Control(pDX, IDC_BUTTON1, m_btn1);

    DDX_Text(pDX, IDC_TIME1, m_ptime1);
    DDX_Text(pDX, IDC_TEMP1, m_ptemp1);
    DDX_Text(pDX, IDC_OXY1, m_poxy1);
    DDX_Text(pDX, IDC_CARBON1, m_pcarbon1);
    DDX_Text(pDX, IDC_ETIL1, m_petil1);
    DDX_Text(pDX, IDC_HUM1, m_phum1);

    DDX_Text(pDX, IDC_TIME2, m_ptime2);
    DDX_Text(pDX, IDC_TEMP2, m_ptemp2);
    DDX_Text(pDX, IDC_OXY2, m_poxy2);
    DDX_Text(pDX, IDC_CARBON2, m_pcarbon2);
    DDX_Text(pDX, IDC_ETIL2, m_petil2);
    DDX_Text(pDX, IDC_HUM2, m_phum2);

    DDX_Text(pDX, IDC_TIME3, m_ptime3);

```



```

        DDX_Text(pDX, IDC_TEMP3, m_ptemp3);
        DDX_Text(pDX, IDC_OXY3, m_poxy3);
        DDX_Text(pDX, IDC_CARBON3, m_pcarbon3);
        DDX_Text(pDX, IDC_ETIL3, m_petil3);
        DDX_Text(pDX, IDC_HUM3, m_phum3);

        DDX_Text(pDX, IDC_TIME4, m_ptime4);
        DDX_Text(pDX, IDC_TEMP4, m_ptemp4);
        DDX_Text(pDX, IDC_OXY4, m_poxy4);
        DDX_Text(pDX, IDC_CARBON4, m_pcarbon4);
        DDX_Text(pDX, IDC_ETIL4, m_petil4);
        DDX_Text(pDX, IDC_HUM4, m_phum4);

        DDX_Text(pDX, IDC_TIME5, m_ptime5);
        DDX_Text(pDX, IDC_TEMP5, m_ptemp5);
        DDX_Text(pDX, IDC_OXY5, m_poxy5);
        DDX_Text(pDX, IDC_CARBON5, m_pcarbon5);
        DDX_Text(pDX, IDC_ETIL5, m_petil5);
        DDX_Text(pDX, IDC_HUM5, m_phum5);

        DDX_Text(pDX, IDC_TIME6, m_ptime6);
        DDX_Text(pDX, IDC_TEMP6, m_ptemp6);
        DDX_Text(pDX, IDC_OXY6, m_poxy6);
        DDX_Text(pDX, IDC_CARBON6, m_pcarbon6);
        DDX_Text(pDX, IDC_ETIL6, m_petil6);
        DDX_Text(pDX, IDC_HUM6, m_phum6);

    //}}AFX_DATA_MAP
}

BOOL CNong2_11View::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs
    return CDaoRecordView::PreCreateWindow(cs);
}

```

```

void CNong2_11View::UpdateResource1()
{
    m_pSet->MoveLast();
    m_ptime1=m_pSet->m_time1;
    m_ptemp1=m_pSet->m_temp1;
    m_poxy1=m_pSet->m_oxy1;
    m_pcarbon1=m_pSet->m_carbon1;
    m_petil1=m_pSet->m_etil1;
    m_phum1=m_pSet->m_hum1;
}

void CNong2_11View::UpdateResource2()
{
    m_p2Set->MoveLast();
    m_ptime2=m_p2Set->m_time2;
    m_ptemp2=m_p2Set->m_temp2;
    m_poxy2=m_p2Set->m_oxy2;
    m_pcarbon2=m_p2Set->m_carbon2;
    m_petil2=m_p2Set->m_etil2;
    m_phum2=m_p2Set->m_hum2;
}

.....

//리소스에 있는 데이터를 모두 m_pSet멤버로 등록한다.
void CNong2_11View::UpdateDb1()
{
    //리소스에 있는 데이터를 View멤버로 옮긴다.
    UpdateData(TRUE);

    //리소스 데이터 -> m_pSet멤버로
    m_pSet->m_time1=m_ptime1;
    m_pSet->m_temp1=m_ptemp1;
    m_pSet->m_oxy1=m_poxy1;
    m_pSet->m_carbon1=m_pcarbon1;
    m_pSet->m_etil1=m_petil1;
    m_pSet->m_hum1=m_phum1;
}

void CNong2_11View::UpdateDb2()

```

```

{
    UpdateData(TRUE);
    m_p2Set->m_time2=m_ptime2;
    m_p2Set->m_temp2=m_ptemp2;
    m_p2Set->m_oxy2=m_poxy2;
    m_p2Set->m_carbon2=m_pcarbon2;
    m_p2Set->m_etil2=m_petil2;
    m_p2Set->m_hum2=m_phum2;
}

```

---

```

void CNong2_11View::OnInitialUpdate()

```

```

{
    //GetDocument()->m_Nong2_11Set;
    m_pSet = new CNong2_11Set(&GetDocument()->m_DB);
    m_p2Set = new CNong2_11Set2(&GetDocument()->m_DB);
    m_p3Set = new CNong2_11Set3(&GetDocument()->m_DB);
    m_p4Set = new CNong2_11Set4(&GetDocument()->m_DB);

    m_ptSet = new CNong2_11Settime(&GetDocument()->m_DB);

    CDaoRecordView::OnInitialUpdate();

    m_btn1.LoadBitmaps(IDB_BITMAP2,IDB_BITMAP2,IDB_BITMAP2)
    m_btn1.SetTextAlignment(CTrackLookButton::AlignRight);

    m_btn2.LoadBitmaps(IDB_BITMAP5,IDB_BITMAP5,IDB_BITMAP5)
    m_btn2.SetTextAlignment(CTrackLookButton::AlignRight);

    m_btn3.LoadBitmaps(IDB_BITMAP4,IDB_BITMAP4,IDB_BITMAP4)
    m_btn3.SetTextAlignment(CTrackLookButton::AlignRight);

    m_btn4.LoadBitmaps(IDB_BITMAP3,IDB_BITMAP3,IDB_BITMAP3);
    m_btn4.SetTextAlignment(CTrackLookButton::AlignRight);

    m_btn5.LoadBitmaps(IDB_BITMAP7,IDB_BITMAP8,IDB_BITMAP7)
    m_btn5.SetTextAlignment(CTrackLookButton::AlignBelow);
}

```

```

        m_btn6.LoadBitmaps(IDB_BITMAP12,IDB_BITMAP12,IDB_BITMAP12)
        m_btn6.SetTextAlignment(CTrackLookButton::AlignBelow);

        if (m_Ports.InitPort(this,1, 9600 ))
            m_Ports.StartMonitoring();
    }

////////////////////////////////////
// CNong2_11View printing

BOOL CNong2_11View::OnPreparePrinting(CPrintInfo* pInfo)
{
    // default preparation
    return DoPreparePrinting(pInfo);
}

void CNong2_11View::OnBeginPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
    // TODO: add extra initialization before printing
}

void CNong2_11View::OnEndPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
    // TODO: add cleanup after printing
}

////////////////////////////////////
// CNong2_11View diagnostics

#ifdef _DEBUG
void CNong2_11View::AssertValid() const
{
    CDaoRecordView::AssertValid();
}

void CNong2_11View::Dump(CDumpContext& dc) const
{

```

```

        CDaoRecordView::Dump(dc);
    }

CNong2_11Doc* CNong2_11View::GetDocument() // non-debug version is inline
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CNong2_11Doc)))
    return (CNong2_11Doc*)m_pDocument;
}
#endif //_DEBUG

/////////////////////////////////////////////////////////////////
// CNong2_11View database support
CDaoRecordset* CNong2_11View::OnGetRecordset()
{
    return m_pSet;
}

/////////////////////////////////////////////////////////////////
// CNong2_11View message handlers

void CNong2_11View::Add1()
{
    m_pSet->AddNew();
    UpdateDb1();
    m_pSet->Update();
    m_pSet->Requery();
    UpdateResource1();
    m_pSet->MoveLast();
}

void CNong2_11View::Add2()
{
    m_p2Set->Open();
    m_p2Set->AddNew();
    UpdateDb2();
    m_p2Set->Update();
    m_p2Set->Requery();
    UpdateResource2();
}

```

```

        m_p2Set->Close();
    }

```

```

LONG CNong2_11View::OnCommunication(UINT ch, LPARAM port)
{

```

```

    unsigned int in;
    double value;
    char tmpbuf[128],tmpbuf1[128];

```

```

        switch(flag_s)

```

```

        {

```

```

            case 0:

```

```

            {

```

```

////////제 1 저장고에 대하여... ////////////

```

```

////////온도 데이터가 들어온시점을 파악하기 위한 부분//////////

```

```

                m_ptime1.Empty();

```

```

                m_ptemp1=0;

```

```

                m_poxy1=0;

```

```

                m_pcarbon1=0;

```

```

                m_petil1=0;

```

```

                m_phum1=0;

```

```

                _strdate(tmpbuf);

```

```

                m_strString=tmpbuf;

```

```

                _strtime(tmpbuf1);

```

```

                m_strString+=" ";

```

```

                m_strString+=tmpbuf1;

```

```

                m_ptime1= m_strString; // time_t defined in <time.h>

```

```

                UpdateData(FALSE);

```

```

////////온도 데이터 처리 부분//////////

```

```

                in=ch;

```

```

                value=TEMP;

```

```

                //m_snumber.Format("%4.2f",value);

```

```

                m_ptemp1=value;

```

```

                UpdateData(FALSE);

```

```

                flag_s++;

```

```

        //flag_p=0;
        Sleep(1000);
        Send1("\xf1");
        break;
    }

////////습도 데이터 처리 부분//////////////////////////////////////
    case 1:
    {
        ///계산식////////////////////////////////
        in=ch;
        value=HUM;
        m_phum1=value;
        UpdateData(FALSE);
        flag_s++;
        Sleep(1000);
        Send1("\xf2");
        break;
    }

////////이산화탄소 데이터 처리 부분//////////////////////////////////////
    case 2:
    {
        in=ch;
        value=CARBON ;
        m_pcarbon1=value;
        UpdateData(FALSE);
        flag_s++;
        Sleep(1000);
        Send1("\xf3");
        break;
    }

////////산소 데이터 처리 부분//////////////////////////////////////
    case 3:
    {
        in=ch;
        value=OXY;
        m_poxy1=value;
        UpdateData(FALSE);

```

```

        flag_s++;
        Sleep(1000);
        Send1("\xf4");
        break;
    }
}
////////에틸렌 데이터 처리 부분////////////////////////////////////////
case 4:
{
    in=ch;
    value=ETIL;
    m_petill=value;
    UpdateData(FALSE);
    Sleep(1000);
    //refresh 요청
    Add1();

    if(addsa2==1)
    {
        flag_s++;
        Send1("\xe0");
    }
    else if (addsa3==1)
    {
        flag_s=flag_s+6;
        Send1("\xf0");
    }
    else if (addsa4==1)
    {
        flag_s=flag_s+11;
        Send1("\xe0");
    }
    else
    {
        SetTimer(3,1000,NULL)
    }
    break;
}
}
////////제 2 저장고에 대하여////////////////////////////////////////
case 5:
{
    //온도 데이터가 들어온시점을 파악하기 위한 부분////////////////////////////////////////
    {
        _strdate(tmpbuf);

```



```

        m_strString=tmpbuf;
        _strtime(tmpbuf1);
        m_strString+=" ";
        m_strString+=tmpbuf1;
        m_ptime2= m_strString; // time_t defined in <time.h>
        UpdateData(FALSE);
////////온도 데이터 처리 부분//////////
        in=ch;
        value=TEMP;
        m_ptemp2=value;
        UpdateData(FALSE);
        flag_s++;
        flag_p++;
        Sleep(1000);
        Send3("\xe1");
        break;
    }
////////습도 데이터 처리 부분//////////
    case 6:
    {
        in=ch;
        value=HUM;
        m_phum2=value;
        UpdateData(FALSE);
        flag_s++;
        Sleep(1000);
        Send3("\xe2");
        break;
    }
////////이산화탄소 데이터 처리 부분//////////
    case 7:
    {
        in=ch;
        value=CARBON;
        m_pcarbon2=value;
        UpdateData(FALSE);
        flag_s++;
        Sleep(1000);
    }

```

```

        Send3("\xe3");
        break;
    }

case 8:
    {
        in=int(ch);
        value=OXY;
        m_poxy2=value;
        UpdateData(FALSE);
        flag_s++;
        Sleep(1000);
        Send3("\xe4");
        break;
    }

case 9:
    {
        in=ch;
        value=ETIL;
        m_petil2=value;
        UpdateData(FALSE);
        //refresh 요청
        Sleep(1000);
        Add2();
        if(addsa3==1)
        {
            flag_s++;
            Send1("\xf0");
        }
        else if (addsa4==1)
        {
            flag_s=flag_s+6;
            Send1("\xe0");
        }
        else
        {
            SetTimer(3,1000,NULL)
        }
        break;
    }
}
.....

```

```

    }

    return TRUE ;
}

BOOL CNong2_11View::OnMove(UINT nIDMoveCommand)
{
switch(nIDMoveCommand)
    {
        //헤더로 이동시
    case ID_RECORD_FIRST:
        //처음 으로 이동
        m_pSet->MoveFirst();
            m_p2Set->Open();
            m_p2Set->MoveFirst();
            m_p2Set->Close();
            m_p3Set->Open();
            m_p3Set->MoveFirst();
            m_p3Set->Close();
            m_p4Set->Open();
            m_p4Set->MoveFirst();
            m_p4Set->Close();
            m_p5Set->Open();
            m_p5Set->MoveFirst();
            m_p5Set->Close();
            m_p6Set->Open();
            m_p6Set->MoveFirst();
            m_p6Set->Close();

        break;
        //마지막으로 이동시
    case ID_RECORD_LAST:
        //마지막으로 이동
        m_pSet->MoveLast();
            m_p2Set->Open();
            m_p2Set->MoveLast();
            m_p2Set->Close();
            m_p3Set->Open();
            m_p3Set->MoveLast();
    }
}

```

```

        m_p3Set->Close();
        m_p4Set->Open();
        m_p4Set->MoveLast();
        m_p4Set->Close();
        m_p5Set->Open();
        m_p5Set->MoveLast();
        m_p5Set->Close();
        m_p6Set->Open();
        m_p6Set->MoveLast();
        m_p6Set->Close();

        break;
case ID_RECORD_NEXT:
    //다음으로 이동시
    m_pSet->MoveNext();
        m_p2Set->Open();
        m_p2Set->MoveNext();
        m_p2Set->Close();
        m_p3Set->Open();
        m_p3Set->MoveNext();
        m_p3Set->Close();
        m_p4Set->Open();
        m_p4Set->MoveNext();
        m_p4Set->Close();
        m_p5Set->Open();
        m_p5Set->MoveNext();
        m_p5Set->Close();
        m_p6Set->Open();
        m_p6Set->MoveNext();
        m_p6Set->Close();

        break;
case ID_RECORD_PREV:
    //이전으로 이동시
    m_pSet->MovePrev();
        m_p2Set->Open();
        m_p2Set->MovePrev();
        m_p2Set->Close();
        m_p3Set->Open();
        m_p3Set->MovePrev();

```

```

        m_p3Set->Close();
        m_p4Set->Open();
        m_p4Set->MovePrev();
        m_p4Set->Close();
        m_p5Set->Open();
        m_p5Set->MovePrev();
        m_p5Set->Close();
        m_p6Set->Open();
        m_p6Set->MovePrev();
        m_p6Set->Close();
        break;
    }
    return CDaoRecordView::OnMove(nIDMoveCommand)
}

```

```

void CNong2_11View::OnStart1()

```

```

{
    char* buf;
    memset(&buf, 0, sizeof(buf));
    flag_s=0;
    if(addsa1==1)
    {
        flag_s=0;
        Sleep(2000);
        Send1("\xf0");
    }
    else if(addsa2==1)
    {
        flag_s=5;
        Sleep(2000);
        Send1("\xe0");
    }
    else if(addsa3==1)
    {
        flag_s=10;
        Sleep(2000);
        Send1("\xf0");
    }
}

```

```

else if(adds4==1)
{
    flag_s=15;
    Sleep(2000);
    Send1("\xe0");
}
}
////////////////////////////////////////보내는 부분////////////////////////////////////////
void CNong2_11View::Send1(char* flag1)
{
    char* bufa;
    memset(&bufa, 0, sizeof(bufa));
    bufa=flag1;
    m_Ports.WriteToPort(bufa);
}

void CNong2_11View::Send2(char* flag2)
{
    char* bufb;
    memset(&bufb, 0, sizeof(bufb));
    bufb=flag2;
    m_Ports.WriteToPort(bufb);
    SetTimer(2,12000,NULL);
}

.....

void CNong2_11View::OnTimer(UINT nIDEvent)
{
    if(nIDEvent==1)
    {
        KillTimer(1);
        Send2("\x44");
    }
    if(nIDEvent==2)
    {
        KillTimer(2);
        Send3("\xf0");
    }
}

```

```

    }

    if(nIDEvent==3)
    {
        KillTimer(3);
        m_ptSet->MoveLast();
        time=m_ptSet->m_dssee;
        if(time==1)
        {
            time=1;
        }
        if(time==2)
        {
            time=2;
        }
        if(time==3)
        {
            time=6;
        }
        if(time==4)
        {
            time=12;
        }
        if(time==5)
        {
            time=24;
        }
        if(time==6)
        {
            time=48;
        }
        int delay;
        Sleep(2000);
        delay = time*3582000;//3600000
        SetTimer(4,delay,NULL);
    }
    if(nIDEvent==4)
    {

```

```

        KillTimer(4);
        OnStart1();
    }
    if(nIDEvent==5)
    {
        char *buff;
        memset(&buff, 0, sizeof(buff));
        KillTimer(5);
        buff="\x44";
        m_Ports.WriteToPort(buff);
        //저장고름들 간의 시간간격.....
        //20초.....
        SetTimer(3,20000,NULL);
    }
    CDaoRecordView::OnTimer(nIDEvent);
}

#include "Sdata.h"

void CNong2_11View::OnData()
{
    // TODO: Add your control notification handler code here
    CSdata *pDlg = new CSdata;
    pDlg->DoModal();
    delete pDlg;
}

//저장고 추가
void CNong2_11View::OnButton6()
{
    COption *pDlg =new COption;
    pDlg->DoModal();
    delete pDlg;
}

void CNong2_11View::OnStrOpt()
{
    CStropt *pDlg =new CStropt;

```



```

        pDlg->DoModal();
        delete pDlg;
    }

void CNong2_11View::OnFileprint()
{
    CString index;
    index.Format("      측정시간      온도      습도      이산화탄소      산소")
    //화면입력에서 -> View멤버로 데이터 얻기
    UpdateData(TRUE);
    //View 데이터를 도큐먼트에 저장
    CNong2_11Doc *pDoc=(CNong2_11Doc *)GetDocument();
    pDoc->m_ptime1=m_ptime1;
    pDoc->m_ptemp1=m_ptemp1;
    pDoc->m_phum1=m_phum1;
    pDoc->m_pcarbon1=m_pcarbon1;
    pDoc->m_poxy1=m_poxy1;
    pDoc->m_petil1=m_petil1;

    pDoc->m_ptime2=m_ptime2;
    pDoc->m_ptemp2=m_ptemp2;
    pDoc->m_phum2=m_phum2;
    pDoc->m_pcarbon2=m_pcarbon2;
    pDoc->m_poxy2=m_poxy2;
    pDoc->m_petil2=m_petil2;

    pDoc->m_ptime3=m_ptime3;
    pDoc->m_ptemp3=m_ptemp3;
    pDoc->m_phum3=m_phum3;
    pDoc->m_pcarbon3=m_pcarbon3;
    pDoc->m_poxy3=m_poxy3;
    pDoc->m_petil3=m_petil3;

    pDoc->m_ptime4=m_ptime4;
    pDoc->m_ptemp4=m_ptemp4;
    pDoc->m_phum4=m_phum4;
    pDoc->m_pcarbon4=m_pcarbon4;
    pDoc->m_poxy4=m_poxy4;
}

```

```

pDoc->m_petil4=m_petil4;

pDoc->m_ptime5=m_ptime5;
pDoc->m_ptemp5=m_ptemp5;
pDoc->m_phum5=m_phum5;
pDoc->m_pcarbon5=m_pcarbon5;
pDoc->m_poxy5=m_poxy5;
pDoc->m_petil5=m_petil5;

pDoc->m_ptime6=m_ptime6;
pDoc->m_ptemp6=m_ptemp6;
pDoc->m_phum6=m_phum6;
pDoc->m_pcarbon6=m_pcarbon6;
pDoc->m_poxy6=m_poxy6;
pDoc->m_petil6=m_petil6;
//프린터 대화상자를 클래스를 설정
CPrintDialog *pDlg = new CPrintDialog(FALSE);
//프린터로 출력시에 출력하는 이름 설정
DOCINFO di={ sizeof(DOCINFO),"Print1: 테스트",NULL};

//대화상자 실행
if(pDlg->DoModal()==IDCANCEL)
{
    delete pDlg;
    return;
}
//프린터 DC를 받는다.
//데이터 출력
HDC hdc= pDlg->GetPrinterDC();
//프린터 DC에 정보를 출력 시작
::StartDoc(hdc,&di);
//페이지 시작을 알린다.
::StartPage(hdc);

::TextOut(hdc,100,100, index, index.GetLength());

::TextOut(hdc,100,250,pDoc->m_ptime1, pDoc->m_ptime1.GetLength())
::TextOut(hdc,700,250,pDoc->m_ptemp1,pDoc->m_ptemp1.GetLength())

```

```

:TextOut(hdc,1000,250,pDoc->m_poxy1,pDoc->m_poxy1.GetLength());
:TextOut(hdc,1300,250,pDoc->m_pcarbon1,pDoc->m_pcarbon1.GetLength());
:TextOut(hdc,1600,250,pDoc->m_petil1,pDoc->m_petil1.GetLength());
:TextOut(hdc,1900,250,pDoc->m_phum1,pDoc->m_phum1.GetLength());

:TextOut(hdc,100,350,pDoc->m_ptime2, pDoc->m_ptime1.GetLength());
:TextOut(hdc,700,350,pDoc->m_ptemp2,pDoc->m_ptemp1.GetLength());
:TextOut(hdc,1000,350,pDoc->m_poxy2,pDoc->m_poxy1.GetLength());
:TextOut(hdc,1300,350,pDoc->m_pcarbon2,pDoc->m_pcarbon1.GetLength());
:TextOut(hdc,1600,350,pDoc->m_petil2,pDoc->m_petil1.GetLength());
:TextOut(hdc,1900,350,pDoc->m_phum2,pDoc->m_phum1.GetLength());

:TextOut(hdc,100,450,pDoc->m_ptime3, pDoc->m_ptime1.GetLength());
:TextOut(hdc,700,450,pDoc->m_ptemp3,pDoc->m_ptemp1.GetLength());
:TextOut(hdc,1000,450,pDoc->m_poxy3,pDoc->m_poxy1.GetLength());
:TextOut(hdc,1300,450,pDoc->m_pcarbon3,pDoc->m_pcarbon1.GetLength());
:TextOut(hdc,1600,450,pDoc->m_petil3,pDoc->m_petil1.GetLength());
:TextOut(hdc,1900,450,pDoc->m_phum3,pDoc->m_phum1.GetLength());

:TextOut(hdc,100,550,pDoc->m_ptime4, pDoc->m_ptime1.GetLength());
:TextOut(hdc,700,550,pDoc->m_ptemp4,pDoc->m_ptemp1.GetLength());
:TextOut(hdc,1000,550,pDoc->m_poxy4,pDoc->m_poxy1.GetLength());
:TextOut(hdc,1300,550,pDoc->m_pcarbon4,pDoc->m_pcarbon1.GetLength());
:TextOut(hdc,1600,550,pDoc->m_petil4,pDoc->m_petil1.GetLength());
:TextOut(hdc,1900,550,pDoc->m_phum4,pDoc->m_phum1.GetLength());

:TextOut(hdc,100,650,pDoc->m_ptime5, pDoc->m_ptime1.GetLength());
:TextOut(hdc,700,650,pDoc->m_ptemp5,pDoc->m_ptemp1.GetLength());
:TextOut(hdc,1000,650,pDoc->m_poxy5,pDoc->m_poxy1.GetLength());
:TextOut(hdc,1300,650,pDoc->m_pcarbon5,pDoc->m_pcarbon1.GetLength());
:TextOut(hdc,1600,650,pDoc->m_petil5,pDoc->m_petil1.GetLength());
:TextOut(hdc,1900,650,pDoc->m_phum5,pDoc->m_phum1.GetLength());

:TextOut(hdc,100,750,pDoc->m_ptime1, pDoc->m_ptime6.GetLength());
:TextOut(hdc,700,750,pDoc->m_ptemp1,pDoc->m_ptemp6.GetLength());
:TextOut(hdc,1000,750,pDoc->m_poxy1,pDoc->m_poxy6.GetLength());
:TextOut(hdc,1300,750,pDoc->m_pcarbon1,pDoc->m_pcarbon6.GetLength());
:TextOut(hdc,1600,750,pDoc->m_petil1,pDoc->m_petil6.GetLength());

```

```

        ::TextOut(hdc,1900,750,pDoc->m_phum1,pDoc->m_phum6.GetLength())
        //DC를 최근의 형태로 재저장한다.
        ::RestoreDC(hdc,-1);
        //페이지 종료를 알린다.
        ::EndPage(hdc);
        //정보출력을 종료 한다.
        ::EndDoc(hdc);
        delete pDlg;
    }

#include "info.h"

void CNong2_11View::OnInfo()
{
    CInfo *pDlg = new CInfo;
    pDlg->DoModal();
    delete pDlg;
}

#include "optgraph.h"
void CNong2_11View::Ongraph()
{
    COptGraph *pDlg = new COptGraph;
    pDlg->DoModal();
    delete pDlg;
}

#include "Addsa.h"
void CNong2_11View::OnAddsa()
{
    // TODO: Add your control notification handler code here

    CAddsa *pDlg = new CAddsa;
    pDlg->DoModal();
    delete pDlg;
}

#include "Datashow.h"

```

```
void CNong2_11View::OnDatashow()
{
    // TODO: Add your control notification handler code here
    CDatashow *pDlg = new CDatashow;
    pDlg->DoModal();
    delete pDlg;
}
```

## 2. 환경관리 시스템용 센서제어기 및 모니터링 하드웨어 개발 프로그램

```
*****  
;DATA      : 2001.02.06  
;CPU       : 8051  
;NAME      ; Mid-Terminal 232  
*****  
          ORG      0  
          JMP      POS  
          ORG     1000  
POS:      NOP  
          NOP  
          MOV     SP,#040H  
          MOV     P1,#000H  
          CALL   DELAY  
          CALL   SINIT  
          MOV     DPTR,#8000H  
          NOP  
  
TOP:      CALL   RRXD  
          MOV     A,#000H  
          MOV     P1,A  
          JMP     TOP  
  
NN:       MOV     R3,A  
          MOV     A,#003H  
          MOVX   @DPTR,A  
          MOV     A,R3  
          CALL   TTXD  
          JMP     TOP  
  
C1:       MOV     A,#002H  
          MOVX   @DPTR,A  
          MOV     A,R3  
          CALL   TTXD  
          JMP     TOP  
  
C2:       MOV     A,#001H  
          MOVX   @DPTR,A  
          MOV     A,R3
```

```

CALL    TTXD
JMP     TOP

C3:
MOV     A,#000H
MOVX    @DPTR,A
MOV     A,R3
CALL    TTXD
JMP     TOP

C4:  MOV     A,R3
      MOV     P1,A

      JMP     TOP

TTXD: JNB     TI,$
      CLR     TI
      MOV     SBUF,A
      RET

RRXD: JNB     RI,$
      MOV     A,SBUF
      CLR     RI
      RET

SINIT: CLR     EA
      MOV     87H,#0
      MOV     TH1,#0FDH

      MOV     SBUF,#0
      RET

DELAY: MOV     R1,#0FFH
      MOV     R7,#0FFH
FOR:   DEC     R7
      DEC     R1
      MOV     R7,#0FFH

      RET
      END

```

```

;*****
;DATA      : 2001.02.26
;CPU       : 8051
;NAME      ; TERMINAL SENSOR
;*****

```

```

        ORG    1000
        NOP
        NOP
CO      EQU    8000H

        MOV    SP,#040H
        NOP
        MOV    P1,#000H
        CALL   DELAY
        MOV    P1,#080H
        CALL   DELAY
        MOV    P1,#000H
        CALL   DELAY
        MOV    P1,#080H
        CALL   DELAY
        MOV    P1,#000H
        CALL   DELAY
        CALL   SINIT
        NOP

TOP:    CALL   RRXD
        MOV    P1, #80H
        CALL   DELAY
        MOV    P1, #00H
        CALL   DELAY

        MOV    DPTR,#CO
        MOV    DPL,A
        MOV    A,#000H
        MOVX   @DPTR,A
        CALL   DELAY
        CALL   DELAY
        MOVX   A,@DPTR

        CALL   TTXD
        MOV    P1, #80H

```



```

        CALL    DELAY
        MOV     P1, #00H
        CALL    DELAY

        JMP     TOP

TTXD:   JNB     TI,$
        CLR     TI
        MOV     SBUF,A
        RET

RRXD:   JNB     RI,$
        MOV     A,SBUF
        CLR     RI
        RET

SINIT:  CLR     EA
        MOV     87H,#0
        MOV     TH1,#0FDH
        MOV     SBUF,#0
        RET

DELAY:  MOV     R1,#0FFH
        MOV     R7,#0FFH
FOR:    DEC     R7
        DEC     R1
        MOV     R7,#0FFH

        RET
        END

```