

최 종
연구보고서

전자동 제재목 소형 건조시스템 개발

Development of Small Scale Lumber Drying System

(주) 건조기술

농 립 부

제 출 문

농림부 장관 귀하

본 보고서를 “전자동 제재목 소형 건조시스템 개발” 과제의
최종보고서로 제출합니다.

2003년 8월 일

주관연구기관명 : (주)건조기술

총괄연구책임자 : 이 형 우

연 구 원 : 최 영 수

연 구 원 : 김 병 남

연 구 조 원 : 김 경 용

요 약 문

I. 제 목

전자동 제재목 소형 건조시스템 개발

II. 연구개발의 목적 및 중요성

목재건조는 목재산업에서 필수적인 공정으로 제품의 품질과 생산성 향상을 위하여 목재 건조 관련 기술의 개선과 개발이 매우 절실히 요구되고 있다. 본 연구에서는 국내 목재산업의 특성에 적합한 전자동 제재목 소형 건조시스템을 개발, 그 성능을 평가, 확인하고자 하였다.

III. 연구개발 내용 및 범위

본 연구에서는 목재건조에 대한 전문적 지식 없이도 건조기를 운영할 수 있는 전자동 제재목 소형 건조시스템을 개발, 보급하고자 하였다. 개인용 컴퓨터를 기반으로 현재 지구 상에서 이용 가능한 수종들에 대한 두께별 적정 건조스케줄을 database화하고, 운영자가 수종과 두께만을 선택, click하면 이후 모든 건조공정이 전자동으로 운영되도록 하였다. 이를 위해서 마이크로파를 이용하는 비파괴 센서를 응용한 건조 중 목재함수율 측정시스템과 온도, 습도 및 풍속 제어시스템 등을 연결하였으며, 건조 중 틀어짐(warp)에 의한 손실을 줄이기 위한 top-load(上部荷重)도 포함되었다.

본 연구에서 개발된 전자동 제재목 소형 건조시스템의 성능을 평가하기 위하여 건조특성, 건조결함, 소요에너지 및 건조비용 등을 분석하였다.

IV. 연구개발 결과 및 활용에 대한 건의

1. 전자동 제재목 소형 건조시스템 개발

목재잔적 규격 1800x1800x4200mm(13.6m³)인 일반형과 규격 2100x2000x7200(30.2m³)인 고온형으로 나누어 개발하였으며, 틀어짐 방지를 위한 상부하중과 공기유동 균일화를 위한 천

공간목 등도 함께 개발하였다.

2. 제어시스템 및 운영 프로그램 개발

Touch-panel형 컴퓨터시스템을 이용, 모든 제어와 계측이 완전 자동화되도록 하였으며, 건조스케줄 database의 구축을 통하여 목재건조 비전문가도 쉽게 운전할 수 있도록 완벽한 user-interface를 실현하였다. 목재의 건조중 함수율을 비파괴적인 방법으로 측정하기 위한 마이크로파 센서를 개발하여 장착하였다.

3. 전자동 제재목 소형 건조시스템의 성능 평가

국산 소나무와 낙엽송을 대상으로 일반 함수율 기준, 시간기준, 연속온도상승 및 습구온도 조절 스케줄 등 4가지 건조스케줄을 개발, 적용 및 성능평가 실험을 실시한 결과 건조결함이 감소되고 건조시간 역시 단축될 수 있었다. 건조시스템의 열효율은 49%에 근접할 정도로 우수하였으며, 에너지비용 역시 저렴하여 목재건조기 시장에서 충분한 경쟁력을 지닌 것으로 판단되었다.

SUMMARY

(영문요약문)

I. PROJECT TITLE

Development of Small Scale Automated Lumber Dryer

II. RESEARCH NECESSITIES AND OBJECTIVES

Lumber drying is one of indispensable processes in wood industry. Improvement and development of drying technologies which can guarantee high efficiency are also required very much. The objectives of this study are developing small scale automated lumber dryers which is suitable for Korea wood industry and testing the performances of these dryers to demonstrate the excellency of them.

III. CONTENTS AND SCOPE OF THE RESEARCH

To achieve the objectives of this project the contents and scopes small scale automated lumber dryer were developed. Touch-panel type computer system which was connected with control and sensor system could control and monitor all the drying process and lumber drying schedule database was embedded into this system for highly-improved user-interface. Microwave-type wood moisture sensor was developed to monitor the wood moisture content nondestructively during drying. Top-load was also developed to reduce drying defects, especially warping. Drying characteristics, drying defects, energy requirements, and drying cost were investigated to demonstrate the performances of these dryers.

IV. RESULTS AND SUGGESTIONS

1. DEVELOPMENT OF SMALL SCALE AUTOMATED LUMBER DRYING SYSTEM

Conventional type small scale lumber drying system with lumber loading room dimension of 1800x1800x4200mm(13.6m³) and high-temperature type with 2100x2000x7200 mm(30.2m³) were developed. Top-load and perforated sticker were also developed to

reduce drying defects and equalize air flow, respectively.

2. DEVELOPMENT OF CONTROL SYSTEM AND OPERATION SOFTWARE

All the control and monitoring operation were fully automated by touch-panel type computer system where wood drying schedule database was embedded. This system could guarantee highly improved user-interface and even non-expert could operate this system. Microwave-type wood moisture sensor was also developed to monitor the wood moisture content nondestructively during drying.

3. INVESTIGATION OF THE PERFORMANCE OF SMALL SCALE AUTOMATED LUMBER DRYING SYSTEM

Conventional kiln drying schedule, time schedule, continuously rising temperature schedule, and drying rate schedule were applied for 50mm-thick Korean red pine and Korean larch to investigate the performance of these drying system. Amount of drying defects could be reduced and drying time also could be shortened. The excellent heat-efficiency of near 49% and lower drying costs of these drying system made it possible to expect the highly competitive power in world market of lumber dryer.

CONTENTS

(영 문 목 차)

CHAPTER 1. INTRODUCTION	10
CHAPTER 2. DEVELOPMENT OF SMALL SCALE AUTOMATED LUMBER DRYER	16
Section 1. Conventional Small Scale Automated Lumber Dryer	17
Section 2. High-Temperature Small Scale Automated Lumber Dryer	20
CHAPTER 3. DEVELOPMENT OF CONTROL AND OPERATION SYSTEM	26
Section 1. Control and Monitoring System	27
Section 2. System Operation Software	32
2.1. Main Window	32
2.2. Lumber Drying Schedule Database	33
CHAPTER 4. PERFORMANCE TESTING METHODS FOR SMALL SCALE AUTOMATED LUMBER DRYER	36
Section 1. Specimens	37
Section 2. Drying Schedules	38
2.1. FPL Conventional Kiln Drying Schedule	41
2.2. Time Schedule	41
2.3. Continuously Rising Temperature Schedule	42
2.4. Drying Rate Schedule	44
Section 3. Drying Characteristics	45
3.1. Moisture Content and Internal Temperature	45
3.2. Drying Defects	45
Section 4. Energy Requirements	48
CHAPTER 5. PERFORMANCE OF SMALL SCALE AUTOMATED LUMBER DRYER	51
Section 1. FPL Conventional Kiln Drying Schedule	52
Section 2. Time Schedule	53
Section 3. Continuously Rising Temperature Schedule	54

Section 4. Drying Rate Schedule	55
Section 5. Drying Rates	56
Section 6. Drying Defects	57
6.1. Warping	57
6.2. Drying Stress	59
6.3. Fixation of Resin	61
Section 7. Heat Efficiency	62
Section 8. Estimation of Drying Costs	63
Section 9. Drying of Lumber for Wood Construction and Cultural Properties	65
LITERATURE CITED	67
CHAPTER 6. SYNTHESIS	69
APPENDIX 1. SOURCE FILES FOR OPERATION SOFTWARE	73
APPENDIX 2. CATALOGUES FOR AUTOMATED LUMBER DRYERS	128

목 차

제 1 장 서 론	10
제 2 장 전자동 제재목 소형 건조시스템 개발	16
제 1 절 일반형 전자동 제재목 소형 건조시스템 개발	17
제 2 절 고온형 전자동 제재목 건조시스템 개발	20
제 3 장 제어시스템 및 운영 프로그램 개발	26
제 1 절 제어·계측시스템 개발	27
제 2 절 시스템 운영 프로그램 개발	32
2.1. Main window	32
2.2. 목재건조스케줄 Database 및 설정 프로그램	33
제 4 장 전자동 제재목 소형 건조시스템 성능 평가 방법	36
제 1 절 공시 제재목	37
제 2 절 적용 건조스케줄	38
2.1. 미국 임산물 연구소 일반 함수율 기준 건조스케줄	41
2.2. 시간 기준 건조스케줄	41
2.3. 연속온도상승 건조스케줄	42
2.4. 습구온도조절 건조스케줄	44
제 3 절 건조특성 측정	45
3.1. 함수율과 목재 내부온도 측정	45
3.2. 건조결함의 측정	45
제 4 절 건조 소요에너지	48
제 5 장 전자동 제재목 소형 건조시스템의 성능	51
제 1 절 일반 열기건조스케줄 적용 건조실험	52
제 2 절 시간 기준 건조스케줄 적용 건조실험	53
제 3 절 연속온도상승 건조스케줄 적용 건조실험	54
제 4 절 습구온도조절 건조스케줄 적용 건조실험	55
제 5 절 건조속도	56

제 6 절 건조결합	57
6.1. 틀어짐	57
6.2. 건조응력	59
6.3. 수지고정	61
제 7 절 열효율	62
제 8 절 건조비용 추정	63
제 9 절 목조건축 및 문화재용 목재의 건조	65
참고문헌	67
제 6 장 종합	69
부 록 1. 운영 프로그램 SOURCE FILE	73
부 록 2. 전자동 목재건조기 CATALOGUE	128

제 1 장
서 론

[기술개발의 필요성]

목재의 건조는 목재가공 공정 중 木製品의 부가가치를 높이기 위한 가장 기본적인 공정이다. 예로부터 전수되어온 건조방법인 천연건조(air-drying)는 시설이나 에너지비용이 거의 들지 않지만 오랜 시간이 소요되고 기후변동이나 악천후에 의한 건조결함이 발생하는 등 심각한 단점이 있다. 특히, 건조결함에 의한 손실액은 총 건조비용 중 가장 큰 비중을 차지하고 있으며, 일 예로 참나무의 천연건조에 의한 건조결함 손실액은 총 건조비용의 13%를 차지한다. 그러나 열기건조(kiln-drying)를 실시할 경우 이를 2% 이하로 낮출 수 있고 건조시간도 상당히 절감할 수 있으며, 최근 들어 생재와 건조재간의 가격차가 점차 확대되고 있어 열기건조의 필요성이 크게 증대되고 있다. 또한, 급격히 증가되고 있는 목조주택, 조경시설물 및 목재문화재용 건축재료의 제작에 있어서도 건조공정을 필수적으로 포함시키려는 추세가 강하게 고조되고 있다.

그러나 목재건조의 가장 일반적인 방법인 열기건조는 목재의 전체 가공공정에 필요한 에너지의 약 60~70%를 차지할 정도로 막대한 에너지가 소요될 뿐만 아니라 기존의 설비로는 소요되는 건조시간도 상당하여 목재가공업체의 자금회전에 커다란 변수로 작용하여 왔다. 또한, 다른 재료들에 비하여 건조가 까다롭고 건조스케줄이라는 독특한 기술을 적용해야 하므로 건조전문가의 채용이 요구된다. 하지만 목재건조 전문가의 수가 근본적으로 부족하고 건조시간이 길어 건조 중 함수율 수준에 따라 건조조건을 수시로 변경해야 하는 목재 건조스케줄의 적용을 위해서는 건조시스템의 자동화가 필수적이다. 그러므로 목재건조 중 건조결함의 발생과 소요에너지 및 시간을 최소화할 수 있는 완벽한 자동화가 실현된 목재건조기의 개발이 가장 뚜렷한 대안으로 부상되고 있다.

전자식 자동제어장치(electronic controller)를 이용, 목재건조기를 자동화함으로써 건조시간 단축에 의한 생산량 증대, 품질의 향상, 인건비 절감 및 에너지의 효율적 이용에 따른 이익 증대 뿐만 아니라 건조공정 중 발생된 정보들을 신속하게 제공하여주고 비교적 정확하게 건조소요시간을 추정할 수 있어 목재의 재고관리도 상당 수준 개선할 수 있으며, 실시간(real time) 일정계획 등 생산관리상의 유연성도 강화할 수 있다. 그리고 얻어진 정보들은 모두 database화하여 이후 경영상 의사결정에 큰 도움을 줄 수 있다. 또한, 작업환경이나 조건이 개선되고 건조설비의 유지와 관리가 수월해지며 오동작시 경고장치 작동으로 돌발적인 사태의 예방도 가능하다.

본 사업에서는 목재건조에 대한 전문적 지식 없이도 건조기를 운영할 수 있는 전자동 제재목 소형 건조시스템을 개발, 보급하고자 하였다. 개인용 컴퓨터를 기반으로 현재 지구상에서 이용 가능한 수종들에 대한 두께별 적정 건조스케줄을 database화하고, 운영자가 수종과 두께만을 선택, click하면 이후 모든 건조공정이 전자동으로 운영되도록 하였다. 이를 위해서 마이크로파를 이용하는 비파괴 센서를 응용한 건조 중 목재함수율 측정시스템과 온

도, 습도 및 풍속 제어시스템 등을 연결하였으며, 건조 중 틀어짐(warp)에 의한 손실을 줄이기 위한 top-load(上部荷重)가 포함되었다.

[국내외 관련기술의 현황]

미국, 유럽, 일본 등 목재산업 선진국들에서는 각 수종에 대한 적정 열기건조스케줄의 개발을 이미 1950년대에 끝내고 보다 건조결합, 시간 및 소요에너지를 절감하는 동시에 특수한 기술이 없는 작업자도 손쉽게 운영할 수 있거나 또는 작업자 없이도 운영이 가능한 목재 건조 자동화시스템 개발에 연구의 초점을 맞추어 왔으며, 이와 같은 자동화에 힘입어 기존의 건조스케줄을 수정, 보완하거나 새로운 개념의 스케줄을 개발하는 작업도 활발히 진행되고 있다.

대표적인 목재건조기 제작업체로는 American Wood Dryers, Inc., Koetter Dry Kiln, Lignomat USA, Ltd., Custom Dry Kiln, Nyle Dry Kiln Systems, Ceaf 등이 있으나 대부분 미국 또는 유럽 기업으로 국내에서 목재건조기를 전문적으로 생산하는 업체는 거의 전무한 실정이다. 그러므로 지금까지 많은 외화를 소비하면서 수입하여 사용해왔던 것이 사실이다. 게다가 수종과 두께에 따른 건조스케줄이 database화되어 대상 목재에 대한 건조스케줄의 기준을 제공해주는 시스템이 개발된 바 없으며, 특히, 국산재 활용도를 높여야 하는 국내 실정에서 외국 기업에게 국산재 건조스케줄의 제공을 바랄 수는 없는 것이다. 본 사업을 통하여 개발될 목재건조기는 수시로 새로운 수종과 두께에 대한 건조스케줄을 up-grade하여 주는 동시에 건조에 대한 자문까지 제공하므로써 사용자의 목재건조에 대한 기술수준을 향상시켜주는 효과까지 기대된다.

한편, 대부분의 건조대행업체들이 인천지역에 집중되어 있어 지리적으로 떨어져 있는 업체에서 건조를 의뢰할 경우 물류비용이 오히려 큰 부담이 되고 건조대행업체 역시 구식의 대용량 건조기를 보유하고 있으므로 소규모 물량에 대해서는 건조를 기피하는 경향이 있어 결국 건조공정이 생략되어 이후 품질에 많은 문제가 도출되는 것이 현 실정이다. 또한, 소량의 특수목을 건조하려는 업체가 늘고 있으나 수종과 규격이 서로 다른 목재를 동시에 건조하므로써 상당한 피해를 입고 있는 것도 사실이다. 그러므로 본 사업에서는 내부용량 24m³(최소 실재적 12m³)의 소형 건조기를 표준 모델로 개발하고자 하였다. 따라서 소규모 기업에게는 공간적 여유를 제공할 수 있고, 중대규모 기업에서도 필요에 따라 건조기의 수를 조절하므로써 경영상 이익을 얻을 수 있을 것이다. 물론 본 기술은 목재건조기의 규모에 관계없이 적용이 가능하므로 주문에 따라 건조기의 규모를 조절할 수 있다.

[기술개발의 효과]

본 기술개발을 통하여 완성된 전자동 제재목 소형 건조시스템의 보급으로 기대되는 효과는 다음과 같이 종합할 수 있다.

- 1) 목제품 품질향상
- 2) 인건비 절감과 효율적 에너지 이용 및 건조시간 단축에 의한 생산성 향상
- 3) 국산재 활용도 개선
- 4) 목재건조기 수입대체 효과 및 제3시장으로 수출 가능
- 5) 목재 재고관리 개선과 생산관리 효율성 강화
- 6) 정보 축적에 따른 기술 수준 향상

[시장현황]

국내에서 목재건조기의 전통적인 사용자로는 우선 약 2,500여개(통계청 광공업통계 1996)의 다양한 규모의 가구제조업체들을 들 수 있다. 근래에는 가구재료의 많은 부분이 MDF나 삭편판과 같은 목질복합재료로 대체되기도 하였으나 최근에는 원목(solid wood)의 사용비율이 높아지면서 다시 목재건조기에 대한 수요가 일고 있다. 게다가 가구산업이 전통적인 중소기업형 산업임을 감안할 때 전국적으로 분포되어 있는 소규모 가구업체에서는 소형 목재건조기가 절실히 요구되고 있다.

그러나 보다 큰 목재건조기 구매자로는 최근 boom이 조성되고 있는 목조주택산업이라 할 수 있다. 또한 조경시설물이나 목조 문화재에 대한 함수율관리가 엄격해지면서 건축용 목재에 대한 건조기 수요가 급증하고 있다. 현재로서는 국내 목조주택업체나 조경건축업체들의 규모가 작은 편으로 역시 소형 건조기에 대한 수요가 충분한 것으로 판단된다. 참고로 국내에는 약 150여개(건설교통부 건설업통계조사보고서 1996)의 조경건축업체가 있다. 하지만 본 기술의 요체는 자동화시스템이므로 건조기의 규모에는 제한이 없어 보다 규모가 큰 업체에 대한 plant 규모의 공급도 가능할 것으로 기대된다. 일 예로 국내 약 1,400여개(산

림청 임업통계연보 1996)의 제재소에서 차별화를 위하여 건조시설과 방부시설을 갖추고자 하는 업체들이 증가하고 있다. 여기에 다양한 목공예업체를 포함한다면 이미 국내에서도 목재건조기 시장의 규모는 비교적 큰 것으로 판단된다.

더구나 거대한 중국시장이 우리 곁에 있다는 사실은 곧 엄청난 기회가 항상 곁에 있음을 의미한다. 그리고 몽고, 러시아, 인도네시아, 말레이시아, 베트남, 캄보디아 등 목재생산국들은 또 다른 거대한 시장이 된다. 실제로 말레이시아와 베트남의 고무나무 생산지에서 목재 건조기에 대한 문의가 접수된 사실로도 국외 시장에 대한 진출은 어렵지 않으며 또한 필히 해야 할 일이다. 그리고 제품의 완성도를 지속적으로 높이면 제품의 가격 대비 성능의 우월성을 무기로 미국, 일본 및 유럽지역으로의 진출 역시 가능할 것으로 기대된다.

표 1. 국내외 목재건조기 시장규모(추정)

구분	항목	업체수	예상주문율(%)	단가(억원)	시장규모(억원)
국내	가구업체	2,500	10	0.5	125
	제재업체	1,400			70
	조경시설업체	150			7.5
	목조주택업체	100			5.0
	합계	4,150			207.5
국외	(국내의 1,000배)	4,150,000	10	0.5	207,500

한편, 우리 나라의 건조기 생산업체는 약 131개(산업자원부.1997.고효율 건조시스템 개발을 위한 기획 연구)가 있으나 목재건조기를 전문적으로 생산하는 기업은 거의 없으며, 일부 생산된 건조기 역시 목재건조에 대한 전문적인 지식이 밀받침되지 못한 향온향습기 수준에 그치고 있다. 따라서 본 전자동 목재건조시스템의 개발에 따른 시장성은 매우 양호한 것으로 판단된다.

[활용방안 및 사업화 계획]

본 기술에 대한 기본적인 사항들은 이미 특허출원 중에 있으며, 본 연구개발 사업을 통하여 축적된 기술들 역시 특허등록으로 산업 권리권을 획득할 예정이다. 그러나 이미 많은 목재관련 기업으로부터 목재건조기 제작에 대한 문의와 협상이 있는 현 상황이 전자동 목재건조시스템의 개발을 기획하게 된 주원인이다.

또한 대부분의 기존 목재건조기가 단순한 향온향습기 수준으로부터 탈피하지 못하고 있는 실정을 감안하여 목재건조스케줄 database의 up-grade를 지속적으로 수행하고, 시스템 구매자에게는 목재건조기술 및 건조시스템 운영에 대한 교육과 함께 A/S의 차원에서 무료로

up-grade를 제공해주므로써 기업과 목재건조시스템에 대한 신뢰를 구축하고자 한다. 이는 결국 미래의 고객을 확보하는 중요한 사업이 될 것이다.

제 2 장

전자동 제재목 소형 건조시스템 개발

제 1 절 일반형 전자동 제재목 소형 건조시스템 개발

건조기는 전체적으로 적정한 단열처리(두께 100mm 우레탄 판넬)를 통하여 열 손실에 의한 에너지 낭비를 억제하였다. 또한, 양순환형 고효율 송풍기를 설계, 4기를 장착 하므로써 건조후 비균일 함수율에 의한 문제를 예방하였다. 한편, 틀어짐에 의한 손실을 억제하기 위해서는 그림 5와 같은 상부하중(top-load)을 설계, 제작하여 적용하였다.

목재의 입출로는 대차(lumber carriage)와 레일(rail)을 이용하여 수행하도록 하였으며, 전기히터(30kW)를 그 열원으로 하였다. 최고온도는 90°C로 하였으며, 습도범위 5 ~ 95%가 유지될 수 있도록 소형 humidifier(5kW)와 배습기(dehumidifier)를 장착하였다. 한편, 건조기 내부 천장에는 천으로 제작하여 승하강이 가능하도록 한 baffle을 설치하여 열풍이 목재 잔적으로 100% 유입되도록 설계하였다.

건조기 내부규격 1800 x 1800 x 4200mm (WHL)로 완성된 일반형 열기건조기 시제품의 구조는 다음의 설계도면 및 사진과 같았다.



그림 1. 일반형 전자동 제재목 소형 건조시스템.

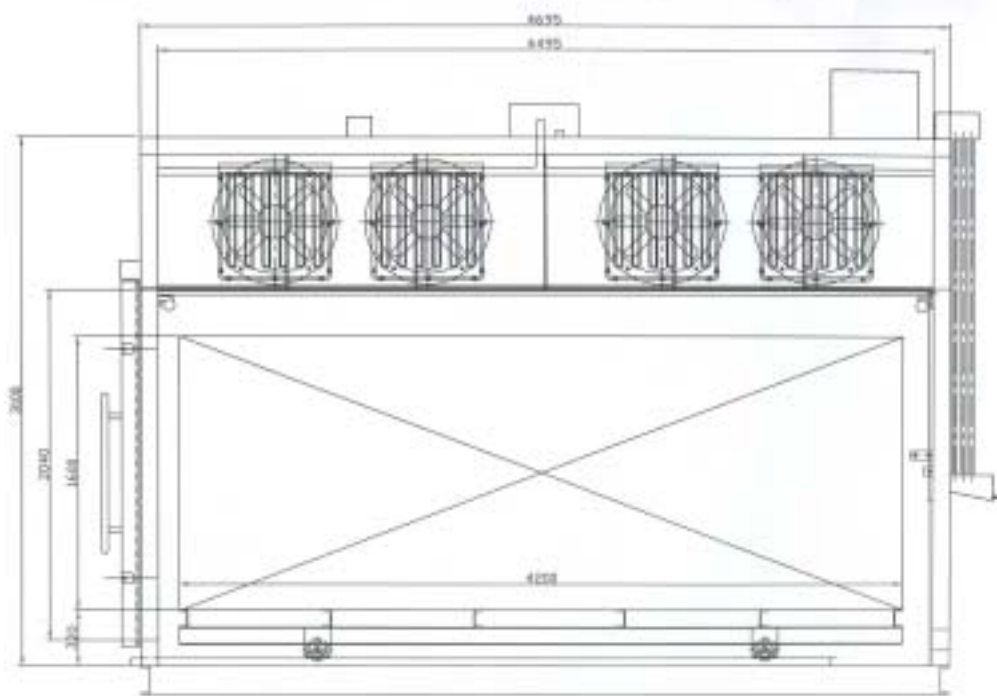


그림 2. 일반형 전자동 제재목 소형 건조시스템 측면도.

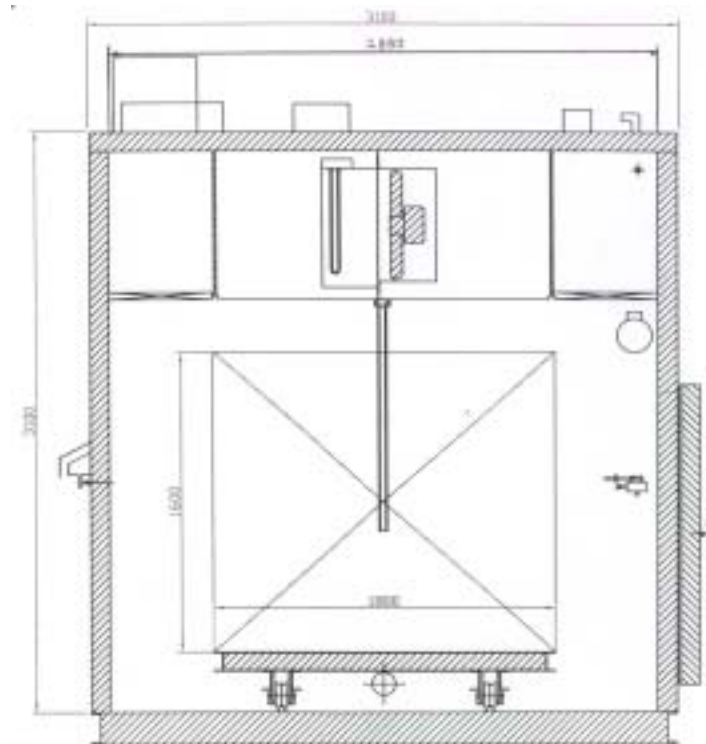


그림 3. 일반형 전자동 제재목 소형 건조시스템 정면도.

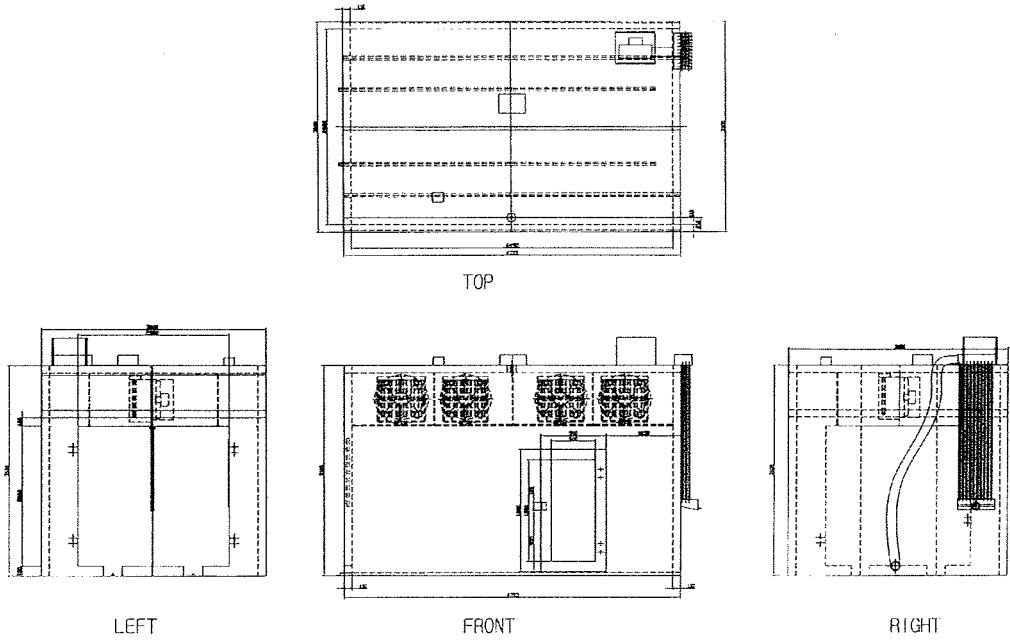


그림 4. 일반형 전자동 제재목 소형 건조시스템 설계도 종합.

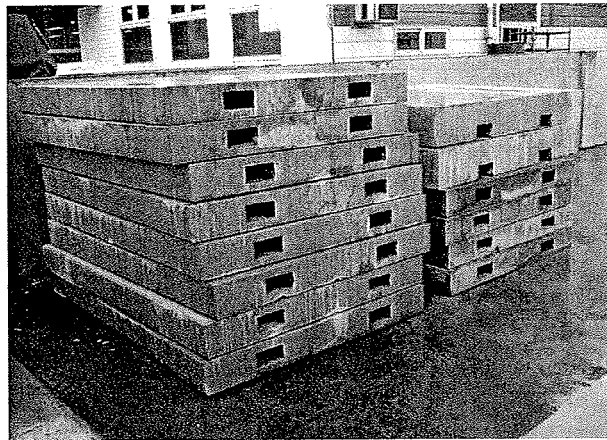


그림 5. 틀어짐 방지용 상부하중(top-load).

제 2 절 고온형 전자동 제재목 건조시스템

고온 건조스케줄을 적용하기 위해서는 완벽한 밀폐성과 견고성이 요구된다. 본 연구에서는 특수 제작된 길이 12m 컨테이너를 이용하여 고온형 전자동 제재목 건조시스템을 제작하였다. 두께 100mm 우레탄 발포로 단열을 하였으며, 건조기 내부의 벽은 부식을 막기 위하여 두께 1.0mm 알루미늄 판을 사용하였다.

제어실을 함께 둔 일체형으로 설계하여 모든 제어와 운전이 건조기에서 함께 이루어지도록 하였다. 따라서 별도의 설치 없이 현장에 건조기를 그대로 놓으면 운전이 가능하며, 이와 같은 설계는 본 건조기의 해외 수출시 작업의 편리성을 도모하기 위함이었다. 목재 잔적의 투입과 건조후 출로는 대차(lumber carriage)와 레일(rail)에 의하여 이루어지도록 하였으며, 일반형과 마찬가지로 상부하중을 두어 틀어짐을 방지하였다.

운전 가능한 최대온도는 150°C이나 안전을 고려하여 120°C 이상에서는 가열매체(50kW)가 운전되지 않도록 하였다. 가습을 위한 소형 humidifier(5kW)와 습기 배출을 위한 배습기가 각각 1대씩 설치되었다. 한편, 1대의 고성능 송풍기는 inverter에 의하여 정해진 시간에 따라 정역회전을 할 수 있도록 하였다.

본 건조기의 설계상 공기의 흐름이 제재목의 섬유방향과 평행하므로 일반형 건조기와는 달리 공기가 유동할 수 있도록 구멍이 뚫린 천공잔목(perforated sticker)을 설계, 제작하였다. 본 천공잔목은 알루미늄을 이용, 두께 25mm로 제작하였으며, 이 천공잔목을 공기의 유동이 가능해질 뿐만 아니라 공기의 흐름을 보다 균일하게 할 수 있는 이점도 확인할 수 있었다.

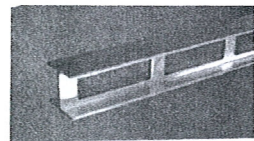
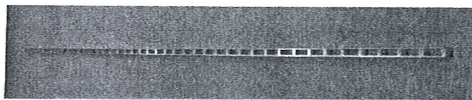


그림 6. 천공잔목.

건조기 내부규격 2100 x 2000 x 7200mm (WHL)로 완성된 본 건조시스템의 설계도는 그림 7과 같으며, 그 제작과정은 그림 8 ~ 11과 같다.

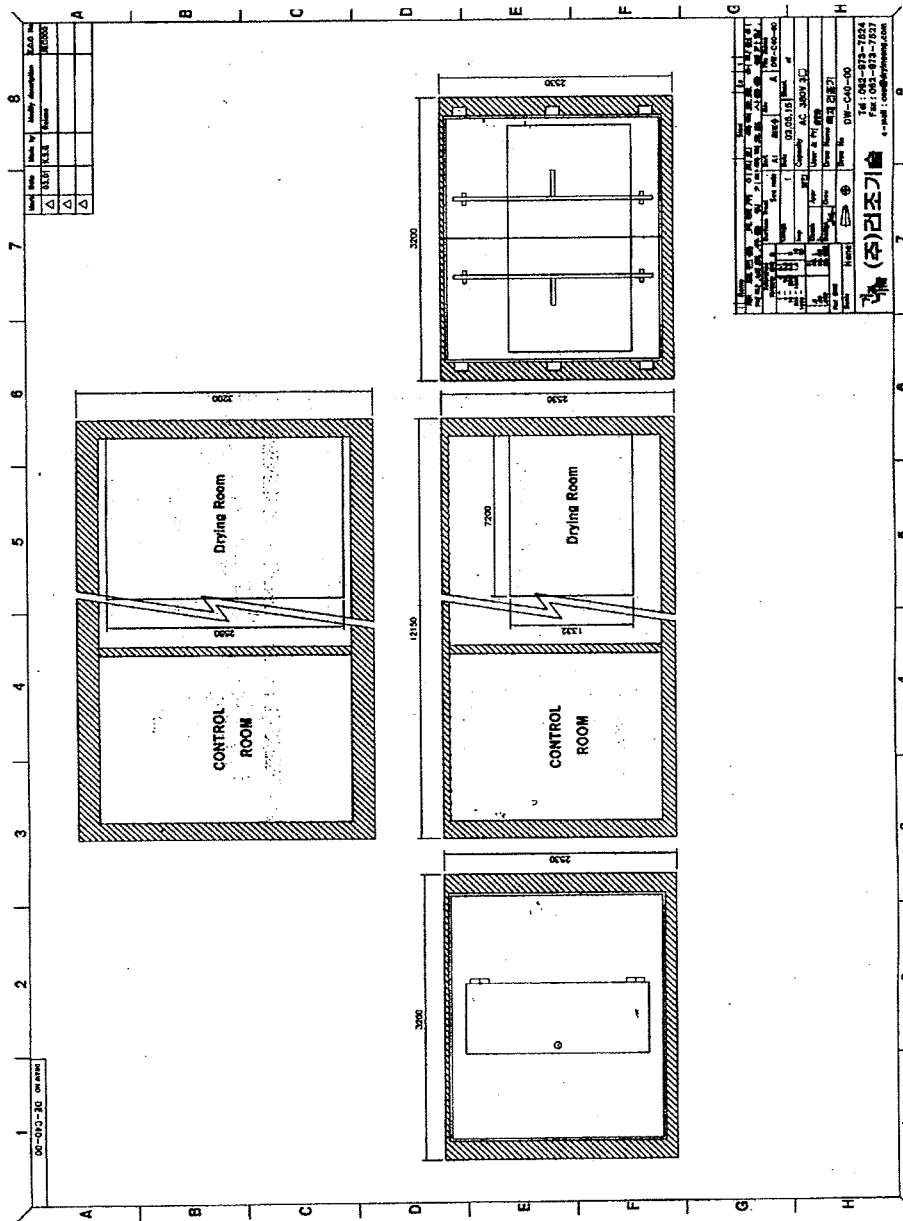


그림 7. 고온형 전자동 제재목 소형 건조시스템 설계도면.

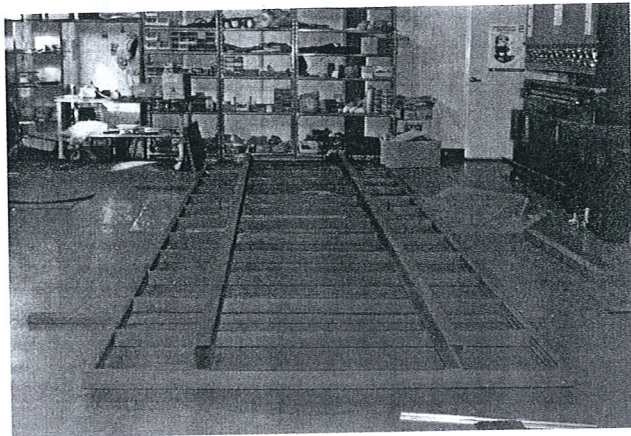
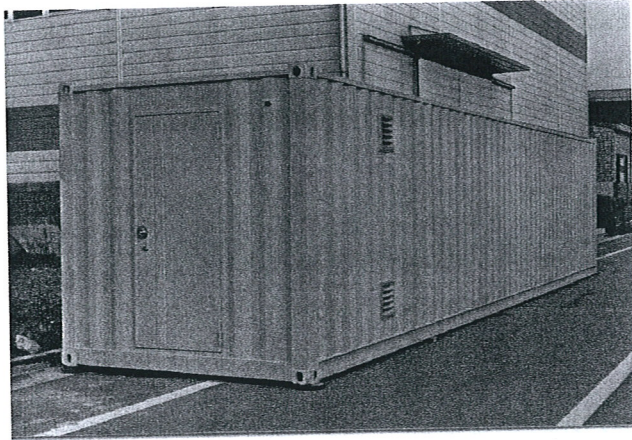
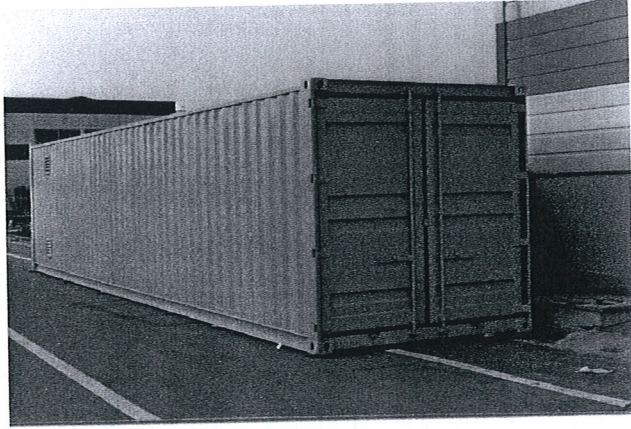


그림 8. 건조기용 컨테이너 입구, 제어실 문 및 대차제작과정.

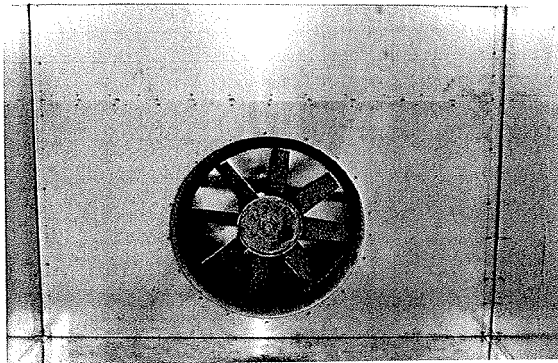
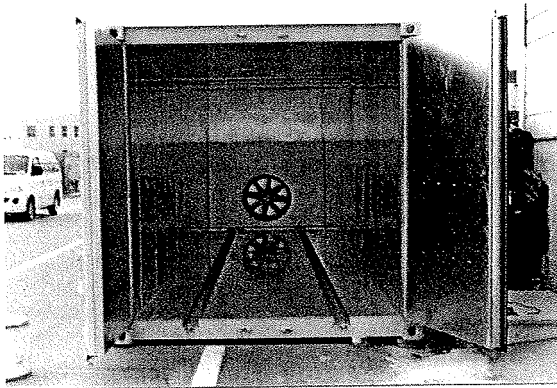
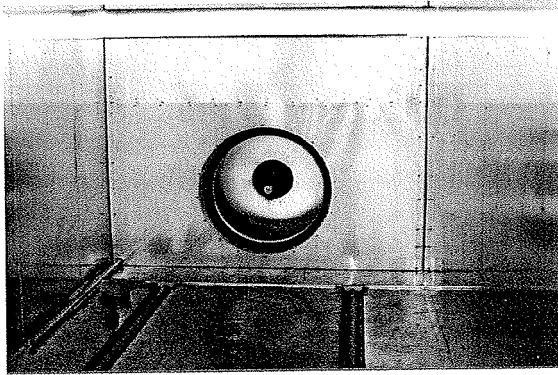


그림 9. 건조기 내부 제작과정 1.

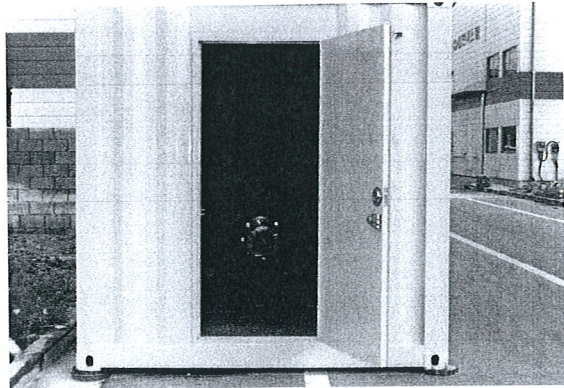
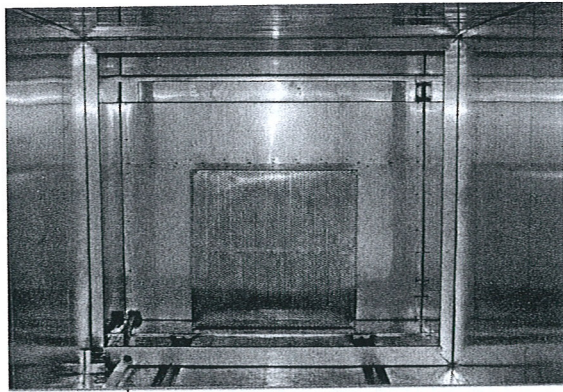
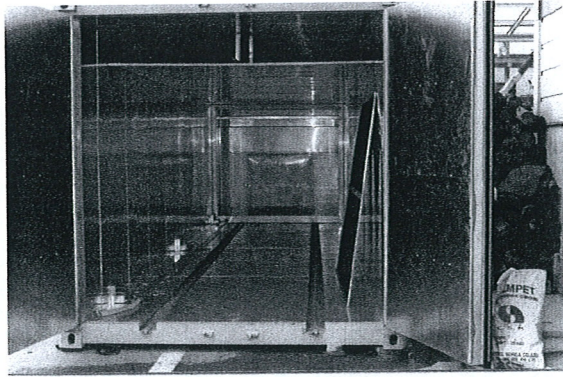


그림 10. 건조기 내부 제작과정 2.

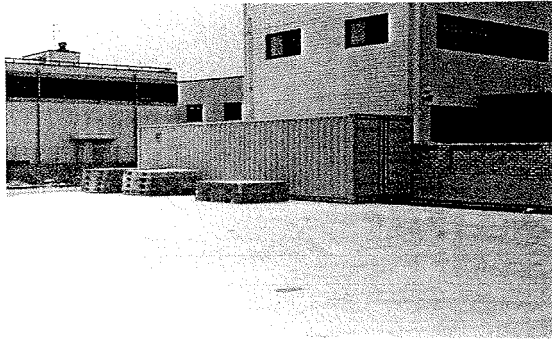
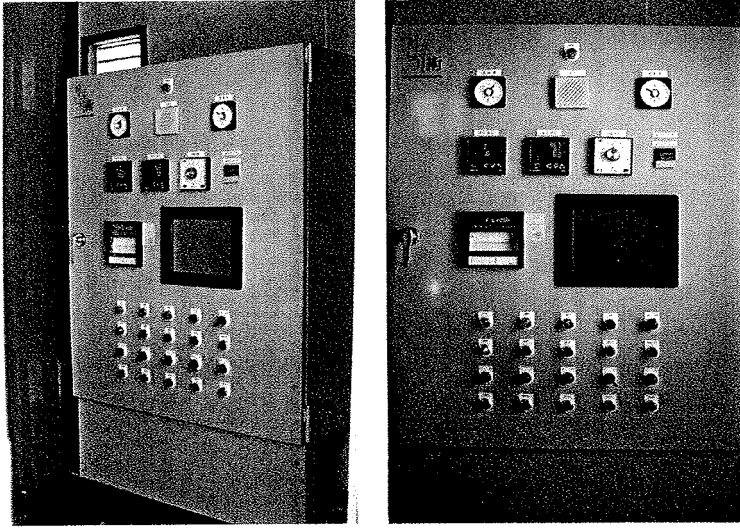


그림 11. Control panel 및 완성된 건조기 모습.

제 3 장

제어시스템 및 운영 프로그램 개발

제 1 절 제어 계측시스템 개발

온도와 습도 등 건조조건을 자동으로 계측 및 제어하는 시스템을 설계하였으며, 건조기 내부의 주요 위치마다 온도센서를 장착하여 건조중 건조기 내부의 건조조건이 균일하게 유지되는지의 여부를 감시할 수 있도록 하였다.

Touch panel을 활용하여 user-interface를 완벽히 편리하게 도모하였으며, 현재 시스템 운영 software를 touch-panel에 loading하여 PID controller 및 inverter 등 제반 제어계측시스템과의 연결을 시도하였으며 완벽히 작동함을 확인하였다.

한편, 본 시스템의 자동 운영을 위하여 핵심이 되는 건조중 함수율 자동계측은 연구초기에는 load cell을 이용하여 건조 중 목재의 중량을 측정, 함수율로 환산하여 측정하였다. 그러나 빠른 풍속과 송풍기에 의한 진동 및 고온고습한 공기조건 등에 의해 심한 noise 가 발생하였다. 따라서 본 연구에서는 마이크로파(200MHz)를 이용한 함수율 센서를 응용하여 목재의 건조중 함수율을 측정하도록 개발, 보정하였다.

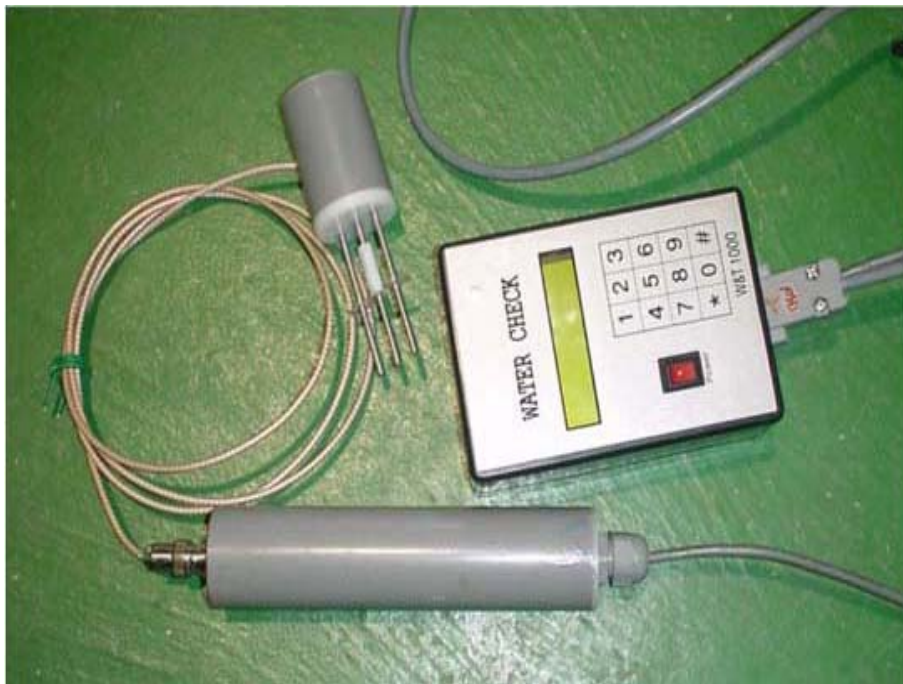


그림 12. 마이크로파 목재 함수율 센서.

마이크로파 함수율 측정기의 보정실험은 국산 소나무, 뉴질랜드산 라디아타소나무, 잣나무, 낙엽송과 상수리나무의 시편을 이용하였으며, 각 수종의 보정결과는 그림 13 ~ 17과 같았다.

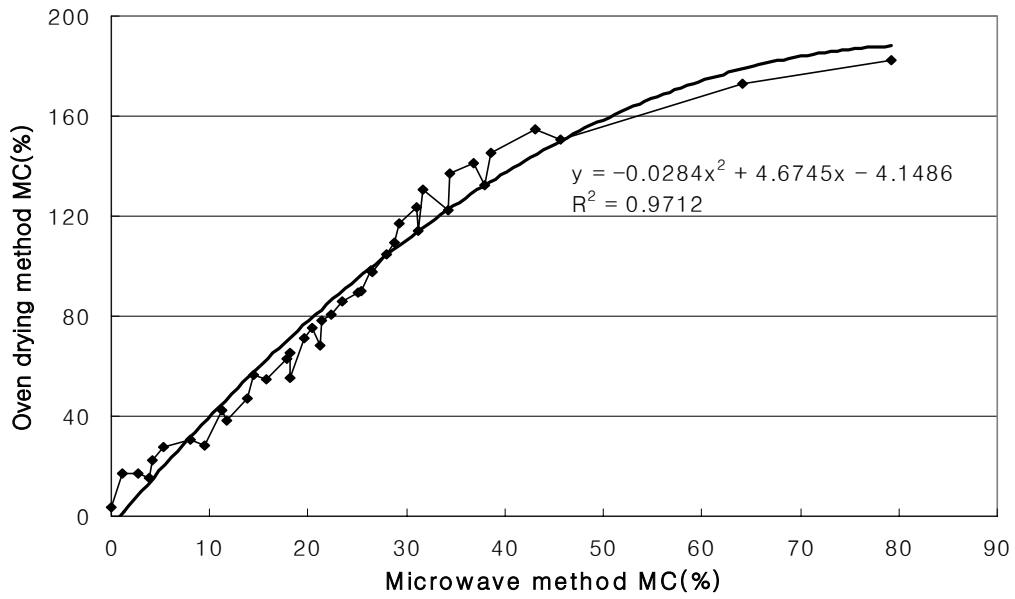


그림 13. 국산 소나무의 마이크로파 함수율 센서 보정곡선.

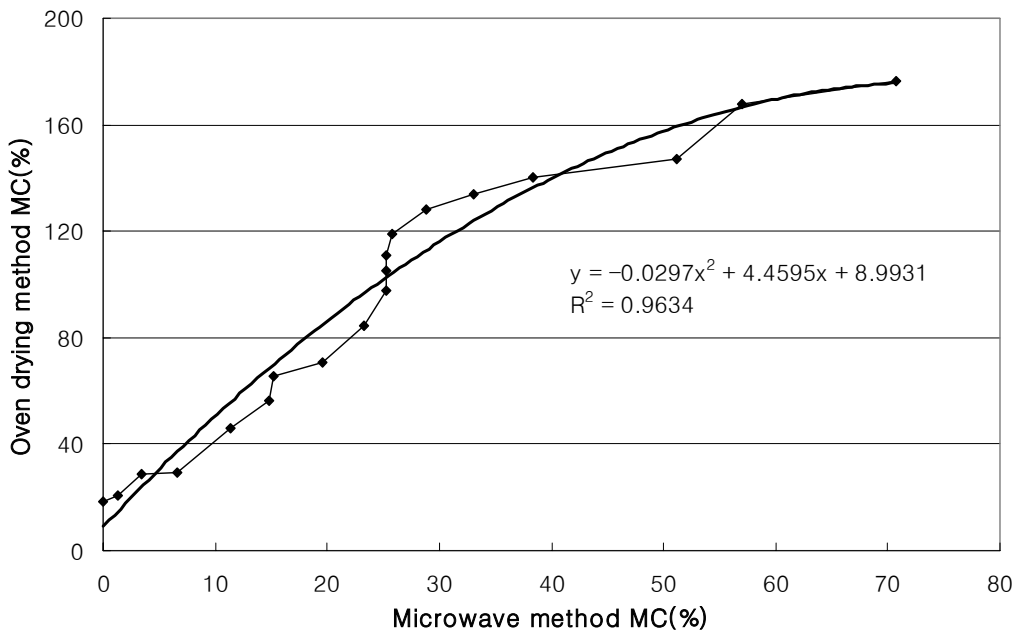


그림 14. 뉴질랜드산 라디아타소나무의 마이크로파 함수율 센서 보정곡선.

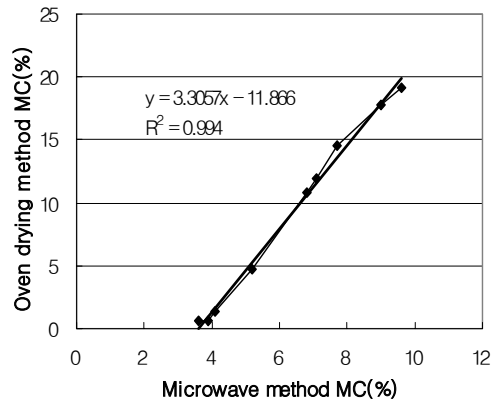
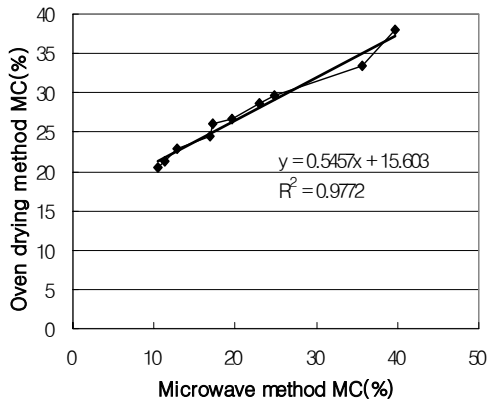
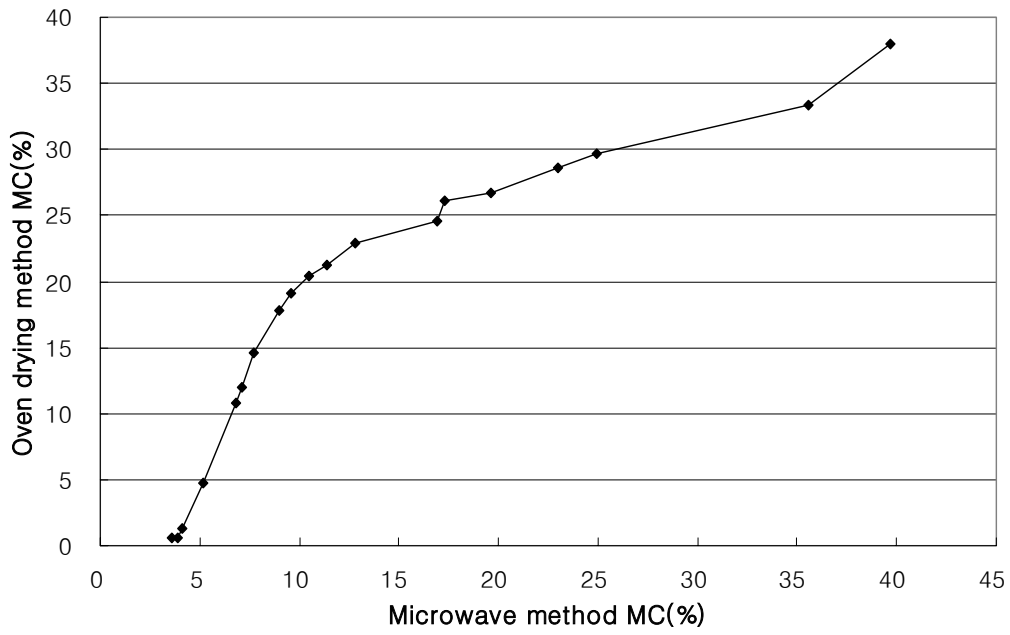


그림 15. 국산 잣나무의 마이크로파 함수율 센서 보정곡선.

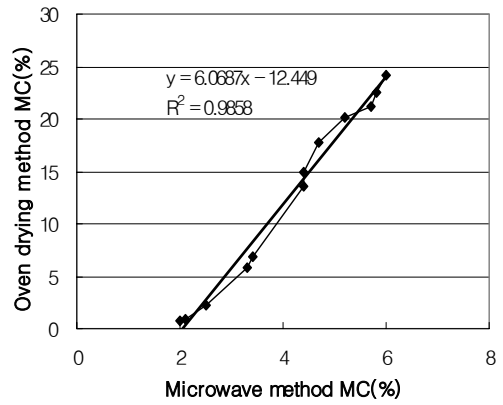
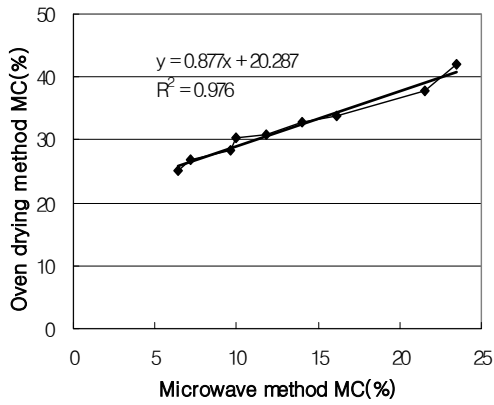
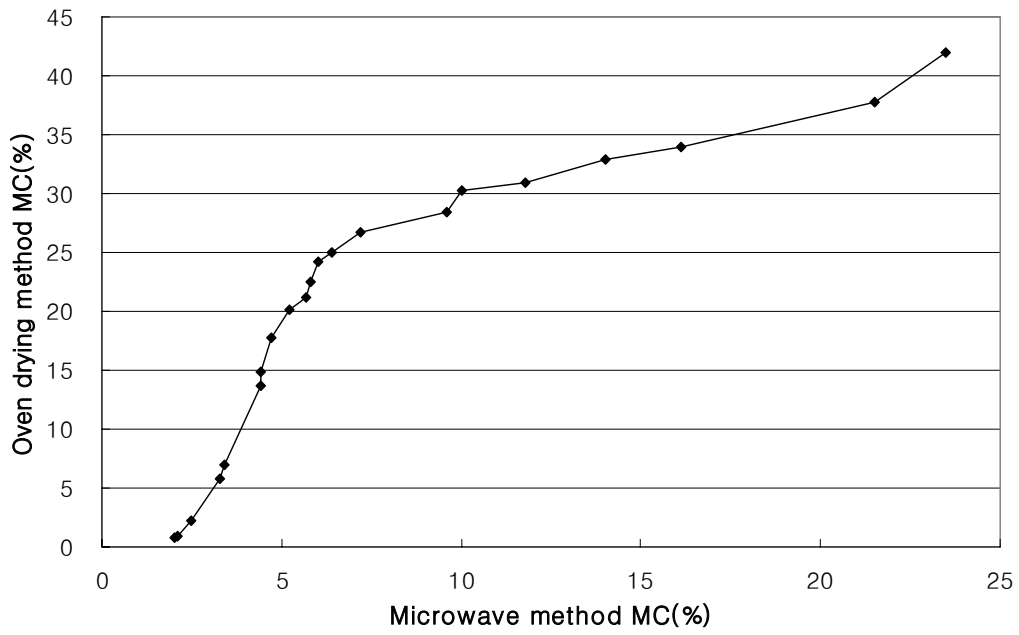


그림 16. 국산 낙엽송의 마이크로파 함수율 센서 보정곡선.

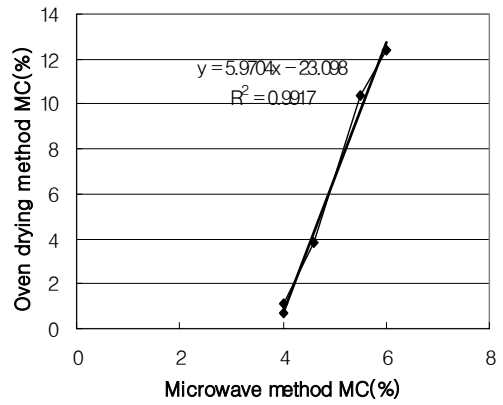
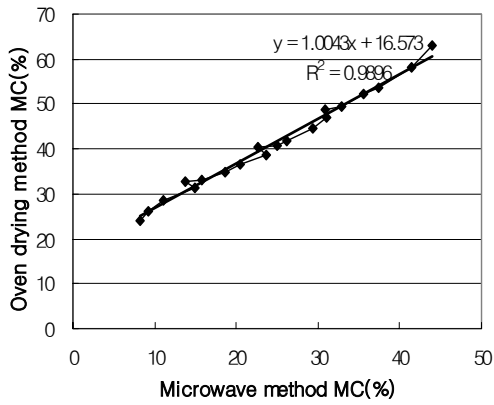
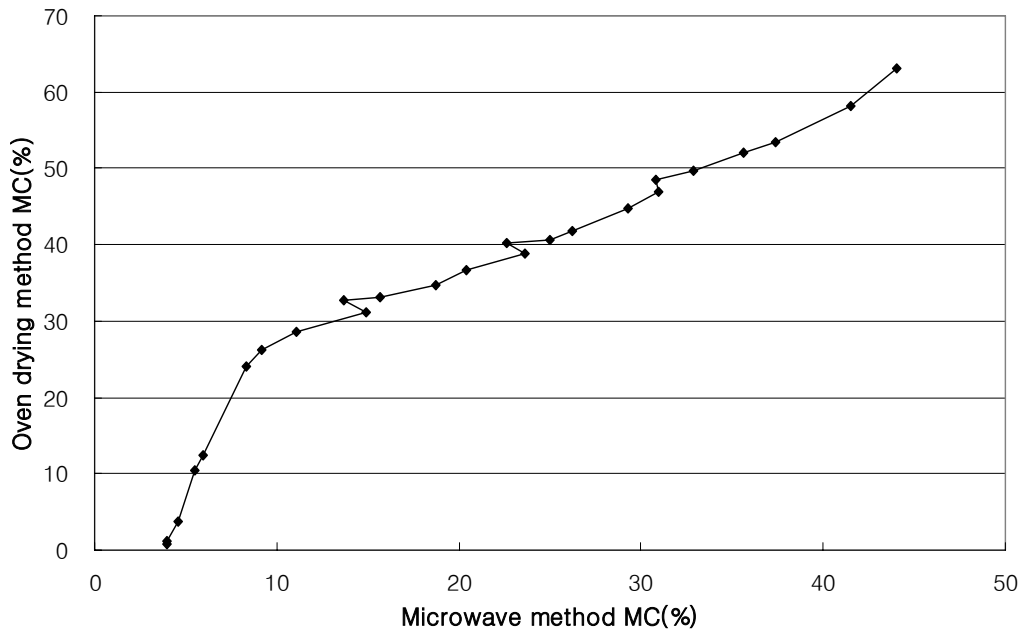


그림 17. 국산 상수리나무의 마이크로파 함수율 센서 보정곡선.

제 2 절 시스템 운영 프로그램의 개발

2.1. Main window

Borland社 C++ Builder를 언어로 하여 시스템 운영 software 개발을 완료하였으며, 작동화면은 다음 그림 18과 같았다.

건조시스템은 미리 설정해둔 우측의 건조스케줄에 따라 자동적으로 운전되며 건조에 관련된 모든 data는 화면 좌측에 표시되도록 하였다. 건조관련 data로는 건습구온도와 건조중 목재의 함수율 및 풍속 등이며, 필요에 따라서는 목재 내부에 온도센서를 삽입하여 건조중 목재내부의 온도를 monitoring 할 수 있도록 하였다. 기타 건습구온도에 따른 상대습도와 평형함수율 등도 표시하였다.

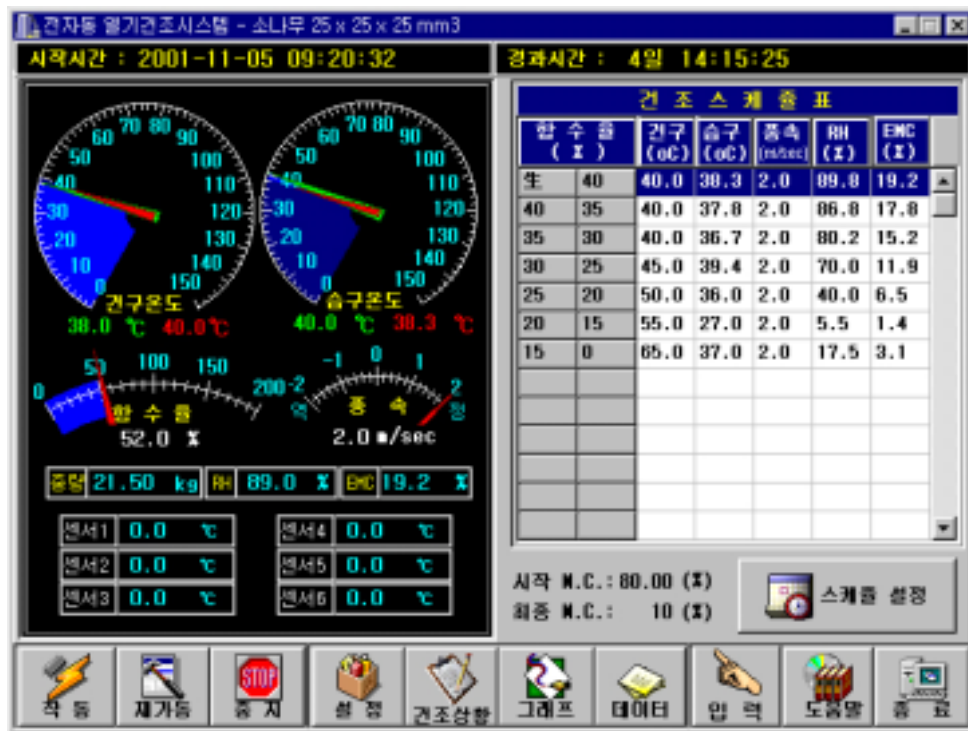


그림 18. Main window.



그림 19. Data monitoring window.

2.2. 목재건조스케줄 Database 및 설정 프로그램

지금까지 전세계적으로 개발되었던 기존의 건조스케줄에 대한 자료를 수집하여 전반적인 database를 구축하였다. 단, 각 스케줄은 크게 국산, 북미제, 열대제 등 산지별로 group 지어 관리가 쉽도록 하였다. 선택된 건조스케줄이 자동적으로 작성되도록 해주는 건조스케줄 작성 프로그램도 개발하였으며, 이때에는 함수율 기준과 시간 기준으로 나누어 개발하였다.

한편, database에 없는 수종이 선택된 경우에도 해당 수종의 비중을 입력하면 건조스케줄을 추정해주는 프로그램도 동시에 개발, 포함시켰으며, 에너지 절감을 위하여 함수율 단계별로 풍속을 제어해주는 프로그램도 개발하였다.

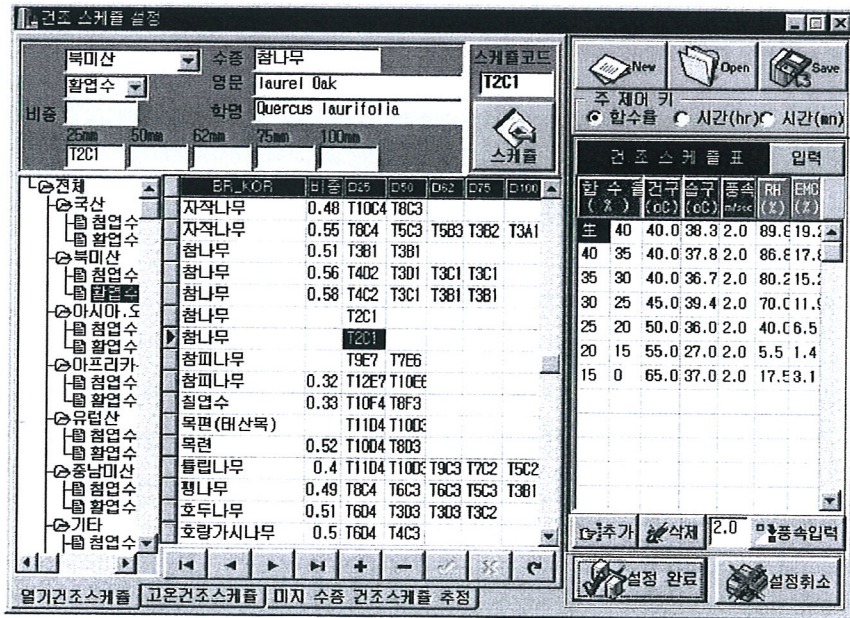


그림 20. 건조스케줄 작성 window.

한편, 그림 21과 22는 본 프로그램을 응용하여 1대의 컴퓨터시스템으로 최대 8기의 건조기를 동시에 제어할 수 있는 프로그램으로 개발된 software의 main window이다.



그림 21. 다중 제어 프로그램 window 1.

The screenshot displays a multi-control program window with several panels:

- Top Panel:** Shows system status including '2' (likely a station or unit number), '수동' (Manual), and '자동' (Automatic) modes. It includes numerical values like '500' and '10.0'.
- Left Panel:** Contains control parameters such as '수동/자동', '수동/자동', and '수동/자동' with associated numerical values and checkboxes.
- Bottom Left Panel:** A table titled '전조스캐온 설정' (Pre-charge Scan Setting) with columns for '시 (hour)', '수', '동', '속', '속', '속', '속'. It contains data for various time intervals.
- Main Panel:** A large data table with columns including '시', '수', '동', '속', '속', '속', '속', '속', '속', '속', '속', '속', '속', '속', '속'. It displays a series of numerical values for different time intervals.
- Bottom Right Panel:** A summary table with three columns labeled '1', '2', and '3', containing various control parameters and their values.

그림 22. 다중 제어 프로그램 window 2.

제 4 장

전자동 제재목 소형 건조시스템 성능 평가 방법

제 1 절 공시 제재목

공시수종은 국산 소나무(*Pinus densiflora* Sieb. et Zucc.), 낙엽송(*Larix kaemferi* Carr)이며, 제재목의 크기, 초기함수율 및 기타 조건은 표 2와 같이 건조스케줄 개발 실험용 제재목의 두께는 25mm, 50mm이고, 폭은 100mm, 150mm 및 270mm이며, 길이는 600mm, 1800mm 및 2400mm의 시편을 사용하였다. 그리고 본 실험은 실험실용 시편(Lab.)과 산업현장용 시편(Industry)으로 구분하여 이루어졌으며, 건조결함 중 틀어짐(warping)의 정도를 비교하기 위하여 소나무와 낙엽송 시편을 이용하여 가장 강한 조건의 건조스케줄 중 하나인 고온건조스케줄을 적용하여 실험하였다.

본 실험에 사용된 제재목 시편의 초기함수율, 규격 그리고 적용 건조스케줄 형식은 다음의 표2와 같다.

표 2. 공시 제재목의 규격과 초기함수율

수종	구분	건조스케줄	규격 (mm)	초기함수율 (%)	
국 산 소나무		일반 함수율 기준 스케줄	25x100x600	155.5	
			50x100x600	101.4	
	실험실 수 준	시간 기준 스케줄	50x100x600	50.2	
			50x100x600	63.8	
			연속온도상승스케줄	25x100x600	132.1
				50x100x600	138.7
	산업용		습구조절스케줄	50x270x1800	169.8
				50x270x1800	150.4
	국 산 소나무	틀어짐 측정용	고온건조스케줄	50x150x2400	31.8
	국 산 낙엽송	틀어짐 측정용	고온건조스케줄	50x150x2400	38.5

제 2 절 적용 건조스케줄

미국, 유럽, 일본 등 목재산업 선진국들에서는 각 수종에 대한 적정열기 건조 스케줄의 개발을 이미 1950년대에 종결하고 건조결합, 시간 및 소요에너지를 보다 절감하는 동시에 특수한 기술이 없는 작업자도 손쉽게 운영할 수 있거나 또는 작업자 없이도 운영이 가능한 목재 건조 자동화 시스템 개발에 연구의 초점을 맞추어 왔다. Wengert 등(1971)이 참나무의 내부할렬 발생을 감소시키기 위해 착안한 무단계형(gradual) 스케줄을 적용하기 위하여 load cell을 이용한 건조자동화시스템의 구성을 시도한 이후 Holmes 등(1976)도 load cell을 이용하여 건조 중 함수율측정을 자동화함으로써 바람직한 결과를 얻은 바 있다.

Dedrick(1968)은 두께 50mm, 폭 200mm인 hemlock을 일반 열기 건조스케줄을 이용하여 건조한 결과 함수율 63.3%에서 13.7%까지 약 120시간이 소요되었다. 따라서 건조속도는 0.41%/hr로 계산된다. 이때 사용된 잔목의 두께는 12mm이었고, 2.75m/sec의 풍속(Reynolds Number = 4,000 (온도 37°C))을 4시간마다 역회전하며 적용하고 있다. 한편, 연속 온도 상승 스케줄 적용 시험에서는 0.9m/sec(Reynolds Number = 1,700 (온도 37.8°C))의 낮은 풍속과 두께 19mm의 잔목을 적용하였다. 공기의 초기 조건은 습도 50%, 온도 32°C이었으며, 1시간당 3.3°C씩 온도를 상승시켜 최종온도는 165.5°C에 도달하였고, 함수율 80%에서 6%까지 40시간만이 소요되었다. 따라서 건조속도는 1.85%/hr로 일반 열기 건조스케줄의 4배 이상에 달함을 보고한 바 있다.

온도가 상승되면 목재는 보다 유연하게 되어 변형(strain)능력이 증가되므로 할렬 발생 가능성을 낮출 수 있으나 동시에 수축률도 증가되어 오히려 할렬 발생 가능성이 높아지기도 한다. 따라서 수축률과 파괴 시 변형량 사이의 관계로부터 최적의 건조조건을 구할 수 있다는 관점에서 시도된 수축-변형 제어형 건조스케줄(SSC : shrinkage-strain control schedule)은 Bello 등(1975)에 의해 최초로 시도된 바 있다. 두께 25mm northern red oak를 대상으로 온도와 함수율에 따른 수축률 및 파괴 시 변형률을 조사, 두 측정량의 교점들을 찾아 가장 낮은 평형 함수율을 나타내는 지점의 평형 함수율과 건조온도를 건조조건으로 결정하는 방법을 제시하였다.

Nassif(1979)가 최초로 시도한 연속 변화형 건조스케줄(CVS ; Continuously varying schedule)이 연속 온도 상승 건조스케줄과 다른 점으로는 건조 중 목재의 함수율 단계에 따라 건조와 습구 온도 모두 상승률을 달리한다는 것이다. 그는 호주산 Eucalyptus에 속하는 두 수종을 대상으로 한 연속 온도 상승 건조스케줄 적용 실험에서 서로 다른 온도 상승 조건으로 실험한 결과 건조 초기에는 낮고 섬유포화점 이하에서는 높은 온도 상승률을 적용하는 것이 바람직하다는 결론을 얻었다. 이와 같은 결과를 바탕으로 건조 초기에는 0.1°C/hr, 그리고 함수율 25% 미만에서는 0.25°C/hr의 온도 상승률(풍속 1.3m/sec, 최고 온도 85°C)을 적용한 결

과 연속 온도상승 건조스케줄과 거의 비슷한 건조속도를 얻은 반면 건조재의 품질을 향상시킬 수 있었으며, 일반 열기건조스케줄(schedule AU)에 비해 건조시간을 1/3 이하(1,305시간 : 412시간)로 단축할 수 있었다.

Shida(1994)는 건조속도를 기준으로 하되 건구온도는 일정 수준에 고정하고 습구온도 즉, 상대습도만을 조절하여 안전한 건조속도를 유지하도록 하는 건조스케줄(DRS : Drying rate schedule)을 제시한 바 있다. 두께와 폭이 모두 120mm인 일본 삼나무 정각재를 대상으로 한 실험에서 건구온도를 80℃에 고정하고 함수율 40% 이상에서는 건조속도 0.3%/hr, 그리고 함수율 40% 이하에서는 건조속도 0.15%/hr가 유지되도록 습구온도를 조절하였다. 건조초기의 건습구온도차는 기존 열기 건조스케줄의 초기조건인 건구온도 60℃와 건습구온도차 4℃에서 얻어지는 평형함수율과 같은 수준이 되도록 건구온도 80℃에서 건습구온도차 3℃(상대습도 88%)를 적용하였다. 한편, 안전 건조속도는 10회 이상의 반복실험을 통해 조건별 할렬발생 여부와 건조속도를 비교하여 결정하였다. 이와 같은 방법의 건조스케줄을 적용한 결과 건조속도(약 150시간)에는 큰 차이가 없었으나 건조재의 품질은 비교적 향상된 것으로 보고 된 바 있다.

1980년대 들어서 급속히 발전된 컴퓨터산업에 힘입어 동시에 8개의 건조실을 제어할 수 있는 산업용 건조자동화시스템이 출현하기에 이르렀으며, Little 등(1986)은 BASIC언어를 사용한 제어프로그램을 작성, 건조 중 함수율 측정에는 전기적 수분계 보다는 load cell을 이용한 방법이 보다 바람직함을 보고한 바 있다. 일본에서도 1980년대 들어서 건조자동화 시스템 개발에 관한 관심이 높아져 Shida 등(1986)도 load cell을 이용한 시스템을 개발하였으며, 유럽에서도 Kunn 등(1988)이 BASIC언어와 자료수집장치(data acquisition system)를 사용한 건조자동화시스템을 구성, 사용자가 보다 쉽게 이용할 수 있도록 window 형태의 메뉴선택방식 등 user-interface기능을 강화한 software를 개발하고자 하였다. 이와 같은 자동화에 힘입어 기존의 건조스케줄을 수정, 보완하거나 새로운 개념의 스케줄을 개발하는 작업도 활발히 진행되고 있다.

대표적인 목재건조기 제작업체로는 American Wood Dryers, Inc., Koetter Dry Kiln, Lignomat USA, Ltd., Custom Dry Kiln, Nyle Dry Kiln Systems, Ceaf 등이 있으나 대부분 미국 또는 유럽 기업으로 국내에서 목재건조기를 전문적으로 생산하는 업체는 거의 전무한 실정이다. 그러므로 지금까지 많은 외화를 소비하면서 수입하여 사용해왔던 것이 사실이다. 게다가 수종과 두께에 따른 건조스케줄이 database화되어 대상 목재에 대한 건조스케줄의 기준을 제공해주는 시스템이 개발된 바 없으며, 특히, 국산재 활용도를 높여야 하는 국내 실정에서 외국 기업에게 국산재 건조스케줄의 제공을 바랄 수는 없는 것이다.

국내에서는 목재 건조시간 단축을 위한 스케줄의 개발뿐만 아니라 개발된 스케줄의 적용을 위하여 필수적인 목재건조 자동화시스템에 관한 연구가 부진한 현실이다. 따라서 목재

산업계에서는 막대한 비용을 들여 미국, 유럽 및 일본 등지에서 시스템을 도입하고 있다. 그러나 전자산업의 발전으로 자동화시스템을 위한 제반 장치들은 국내에서 합리적인 가격에 모두 구입할 수 있는 실정임을 감안할 때 이와 같은 문제의 원인은 hardware를 작동시킬 수 있는 software개발에 대한 관심부족과 자동화를 실현하겠다는 목재기업인의 의지결핍으로 귀결된다고 하겠다. 그러나 software는 유능한 전기, 전자공학자나 프로그래머라 하더라도 그들이 독자적으로 개발할 수는 없는 것으로, 목재건조 자동화를 위한 software개발의 가장 기본적이고도 중요한 것은 바로 목재와 목재건조에 관한 지식과 경험으로 그들과 목재건조 전문가들이 함께 연구할 때만이 개발이 가능하다고 하겠다. 그리고 앞으로 자동화시스템 발전방향은 보다 효율적이고 사용이 간편한 다기능 software의 개발에 초점이 맞추어질 것이 분명하다.

본 연구에서는 다양한 건조스케줄들을 본 연구에서 개발된 전자동 건조시스템에 적용하여 그 성능을 평가하고자 하였다.

2.1. 미국 임산물연구소 일반 함수율 기준 스케줄

(FPL schedule; Conventional kiln drying schedule)

미국 임산물연구소(Forest Products Laboratory)에서는 기호를 사용하여 건조스케줄을 보다 쉽게 작성할 수 있는 방법을 개발한 바 있다. 스케줄 번호(schedule code)는 온도를 의미하는 기호(T1~T14)와 건습구온도차를 의미하는 기호(A~F와 1~8)의 조합으로 이루어진다.

본 실험에서는 연속 온도상승 스케줄 개발의 전 단계로써 국산 소나무(25×100×600mm, 50×100×600mm)를 실험실용 시편으로 제작하여 일반적으로 이용되고 있는 미국 임산물연구소의 스케줄 중에 T8F4를 적용하였다. 이 스케줄의 자세한 건·습구온도, 건습구온도차와 상대습도의 내용은 표 3과 같았다.

표 3. 일반 함수율 기준 스케줄 (schedule code : T8F4)

함수율 (%)	건구온도 (°C)	습구온도 (°C)	건습구온도차 (°C)	상대습도 (%)
생재 - 70	55	51.1	3.9	80.5
70 - 60	55	49.4	5.6	73.3
60 - 50	55	46.7	8.3	62.3
50 - 40	55	44.0	11.0	52.5
40 - 35	55	41.0	14.0	42.5
35 - 30	55	38.0	17.0	34.2
30 - 25	60	40.0	20.0	30.0
25 - 20	65	45.0	20.0	32.5
20 - 15	70	50.0	20.0	35.0
15 - 최종	80	52.0	20.0	24.0

2.2. 시간 기준 건조스케줄(Time schedule)

북미에서는 침엽수의 건조에 대부분 시간 기준 건조스케줄을 사용하고 있다. 건조조건은 12시간 또는 24시간 단위로 변경되는 것이 보통이다. 시간 기준 건조스케줄은 특히, 건조장치의 성능과 풍속 등에 의한 영향을 크게 받으므로 이미 개발된 스케줄이라 할지라도 사용자의 조건에 따라서 변경되어야 한다. 일 예로 미국 임산물 연구소에서 제시하고 있는

시간 기준 건조스케줄은 최소 풍속 2m/sec를 기준으로 하고 있다. 그러므로 시간 기준 건조스케줄의 적용에는 많은 경험과 자료가 축적된 후에 가장 최적의 건조스케줄을 자체적으로 얻어내는 것이 바람직하다.

본 실험에서는 앞서 이루어진 일반열기건조스케줄과 연속 온도상승스케줄 적용실험의 결과를 토대로 산업현장에서 건조공정 작업자가 건조스케줄을 적용함에 있어서 편리성을 감안하여 건조경과시간에 따라 건조기의 건구온도와 습구온도의 제어가 이루어지는 개념의 시간 기준 건조스케줄 적용실험을 실시하였다.

본 실험에서 적용된 시간 기준 건조스케줄은 10시간 단위로 건구온도와 습구온도를 조절하였고, 국산 소나무(50×100×600mm)를 대상으로 한 건조실험에 적용된 시간 기준 건조스케줄의 구성은 표 4와 같았다.

표 4. 두께 50mm 국산 소나무용 시간 기준 스케줄

시간 (hr)	건구온도 (°C)	습구온도 (°C)
0 - 10	74	67
10 - 20	76	68
20 - 30	78	68
30 - End	82	68

2.3. 연속 온도상승 건조스케줄

(CRT; continuously rising temperature schedule)

건조 중 목재를 둘러싼 공기의 온도가 비교적 낮더라도 시스템 내에서 공기와 목재표면간의 온도경사가 열의 유동을 발생시키기에 충분하다면 목재로부터 수분을 증발시킬 수 있고, 에너지의 효율도 크게 향상시킬 수 있다. 따라서 제한된 공간내의 공기온도를 일정한 온도경사가 유지되도록 계속 상승시킨다면 건조속도는 목재로의 열 유입속도에 비례하므로 건조속도와 함께 에너지 소비량도 현격히 감소시킬 수 있을 것이다.

건조 초기에는 목재의 표면이 급속히 건조되면 건조능력에 의한 결함 발생 가능성이 클 뿐만 아니라 열전달도 불량해진다. 따라서 일반적인 건조방법에서는 건조초기 높은 상대습도가 필요하게 되며, 높은 열전달을 위하여 풍속을 증가시켜 난류(turbulent flow)를 형성하는 것이 일반적이다.

그러나, 풍속을 낮추어(0.5 ~ 1.0 m/sec) 목재표면을 층류(laminar flow)를 유지하는 얇은 공기층이 둘러싸도록 한다면 목재로부터 증발된 수분은 이 공기층에 흡수되어 높은 상대습도를 유지하게 되고 그 외의 공기층은 높은 건습구온도차를 갖게되어 열유입 향상효과와 건조능력 완화효과를 동시에 기대할 수 있을 것이다. 즉, 공기의 정밀한 습도조절이 필요 없게 되는 동시에 저풍속으로 인한 전기에너지 절감의 효과도 기대할 수 있다. 또한 비균일 함수율 예방을 위하여 송풍의 회전을 주기적으로 역전해야 하는 수고도 덜 수 있다. 단, 주기적인 습기제거는 필요할 것으로 판단된다.

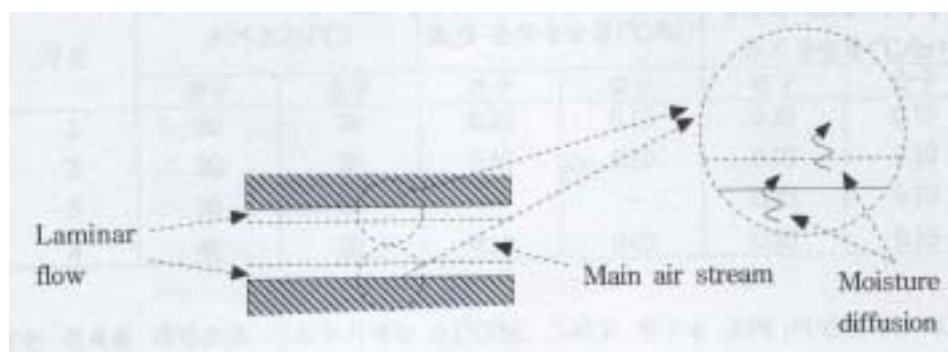


그림 23. 목재건조중 층류형성과 수분확산.

본 실험에서는 미국 임산물연구소 건조스케줄을 적용한 건조실험 결과를 기본으로 하여 국산 소나무재의 연속 온도상승 건조스케줄을 개발하기 위하여 시편의 두께와 적용방법을 달리하여 건조실험을 하였다.

먼저 두께 25mm 국산 소나무 시편의 경우에는 두께가 얇은 것을 감안하여 초기 건구온도를 30℃부터 시작하여 시간당 1.25℃씩 상승시켰고, 습구온도의 경우는 26℃에서 시작하여 시간당 0.65℃씩 상승시켰다. 그리고 전체 건조시간을 40시간으로 하고 최종 건구온도는 80℃, 최종 습구온도는 52℃에서 건조를 완료하였다.

두께 50mm 국산 소나무 시편의 연속 온도상승 건조스케줄은 건조초기 생재상태에서 함수율 25%까지 건구온도를 55℃로 유지하고 함수율 25%이하에서는 시간당 0.6℃씩 증가시켜서 최종 건구온도인 90℃에서 온도상승을 중단하도록 하였다. 그리고 건조기 내의 상대습도를 조절하는 방법을 택하였는데 건조초기에는 상대습도 80.5%에서 시작하여 함수율 40%까지는 시간당 상대습도를 0.5%씩 감소시켰고, 할렬발생 가능성이 높은 함수율 40~25%까지는 상대습도를 55.5%로 유지시켰으며, 함수율 25%부터는 시간당 0.5%씩 상대습도를 감소시켜서 최종상대습도를 22%가 되게 하였다.

두께별 국산 소나무 시편에 적용한 연속 온도상승 건조스케줄은 표 5와 표6과 같았다.

표 5. 두께 25mm 국산 소나무용 연속온도상승스케줄

건구온도 (℃)	습구온도 (℃)	최종 건구온도 (℃)	최종 습구온도 (℃)	온도상승시간 (hr)
30+1.25△T/hr	26+0.65△T/hr	80	52	40

표 6. 두께 50mm 국산 소나무용 연속온도상승스케줄

건구온도 (℃)		상대습도 (%)	
초기	55	초기	80.5
최종	90	최종	22
생재 - 40%	55	생재 - 40%	-0.5△%/hr
40 - 25%	55	40 - 25%	55.5%
25 - 최종	0.6△T/hr	25 - 최종	-0.5△%/hr

2.4. 습구온도조절 건조스케줄(DRS : Drying rate schedule)

본 실험에서는 산업현장에서 직접 사용이 가능한 전자동 제재목 소형 건조시스템을 제작하여 그 성능 테스트와 함께 새로운 개념의 건조스케줄을 개발하기 위하여 산업현장에서 사용되는 구조재의 규격으로 시편을 제작하였고, 함수율 변화에 따라 건구온도는 일반 건조스케줄의 최고온도인 80℃로 고정한 채로 습구온도를 조절하여 건조기 내의 상대습도를 감소시켜 비교적 균일한 건조속도로 목재를 건조한 후 목재의 건조결함율 측정함으로써 건조시스템과 습구온도조절 건조스케줄의 적용여부를 알아보고자 하였다.

본 연구에서는 우리 전통 가옥의 마루판용으로 사용되는 50×270×1800mm 규격의 국산 소나무를 대상으로 실험을 하였으며, 이 실험에서 적용된 습구온도조절 건조스케줄은 표 7과 같았다.

표 7. 규격 50×270×1800mm 국산 소나무의 습구온도조절스케줄

함수율 (%)	건구온도 (℃)	습구온도 (℃)
생재 - 100	80	73
100 - 50	80	63
50 - 25	80	63
25 - 15	80	58
15 - 최종	냉각	냉각

제 3 절 건조특성 측정

3.1. 함수율과 목재내부 온도측정

건조 중 목재 내부 온도 측정은 시편의 가운데 부분에 직경 약 2mm의 구멍을 뚫고 온도센서를 삽입한 후에 thermal grease로 마감하여 구멍내부로 외부공기가 유입되는 것을 차단함으로써 외부로부터의 열전달을 최소화하였다.

건조 실험 중에 함수율 측정을 위한 마이크로파 함수율 측정기와 내부온도 측정을 위한 온도센서가 삽입된 모습은 그림 24와 같았다.

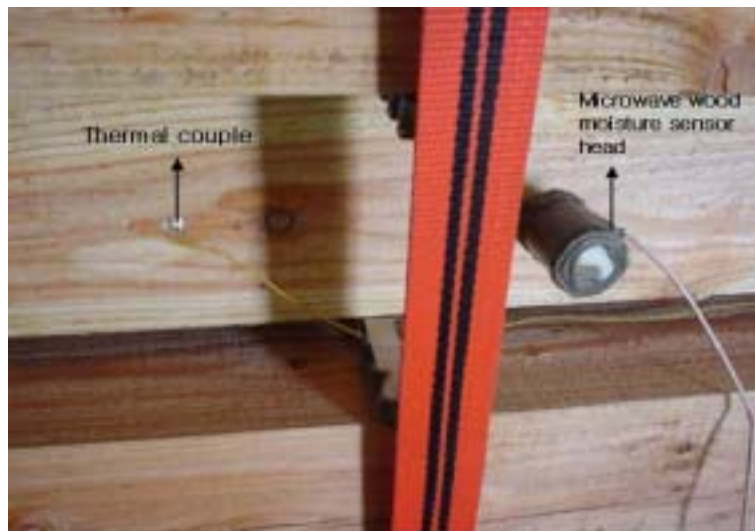


그림 24. 마이크로파 함수율 센서와 온도센서 삽입 모습.

3.2. 건조결함의 측정

1) 틀어짐 측정

건조가 완료된 후에 시편들 각각의 틀어짐(warping)의 치수를 측정함으로써 건조결함의 정도를 알아보았다. 건조 후 각 시편의 길이방향과 측면방향으로 가장 많이 굽은 부분의 높이를 측정하여 각각의 길이굽음(bow)과 측면굽음(crook)을 구하였다. 비틀림(twist)은 한 쪽 끝을 가볍게 누르고 나머지 한쪽이 바닥에서 뜬 정도를 측정하여 비틀림을 구하였다. 그리고 비슷한 방법으로 시편의 폭방향 변형 정도를 측정하여 너비굽음(cup)을 측정하였다.

본 실험에서는 건조 중 상부하중(top-load)의 효과를 알아보려고 상부하중 처리를 한 시

편과 처리를 하지 않은 시편으로 각각 나누어서 같은 건조조건으로 건조를 완료한 후 틀어짐의 치수를 측정하였다. 그리고 틀어짐 측정을 위해 사용된 시편은 국산 소나무와 낙엽송으로 시편의 규격은 50×100×2400mm이었고, 이때 적용된 건조스케줄은 목재가 가장 잘 틀어지기 쉽다고 여겨지는 고온건조스케줄을 적용하였으며 상부하중 압력 500kgf/cm²이 적용되었다. 본 실험에서 틀어짐 측정을 위해 적용한 고온건조스케줄은 표 8과 같았다.

표 8. 두께 50mm 국산 소나무와 낙엽송용 고온건조스케줄

함수율 (%)	건구온도 (°C)	습구온도 (°C)	건습구온도차 (°C)
Green - 100	95	95	0
100 - 70	100	95	5
70 - 40	105	95	10
40 - 20	110	95	15
20 - End	110 → 40	95 → 33	15 → 7

2) 건조응력 측정

표면경화(casehardening)의 원인은 너무 급속한 건조나 불균일 건조에 기인한다. 건조실내 고온저습 또는 온도와 습도의 변경이 급할 때 표면건조는 빨리 된다. 그래서 건조초기 목재는 표층에 인장응력, 내층에 압축응력이 나타난 상태를 역표면경화(reverse casehardening)라 부른다. 이와 같은 표면경화 전기의 목재를 거단하면 외측으로 굽는다. 이것을 방지해 두면 내부가 건조되면서 정상적인 수축을 하게 되므로 방향이 바뀌면서 내측으로 굽는다. 표면경화 전기에서 건조가 진행됨에 따라 내층은 인장응력, 외측은 압축응력이 나타내는 상태를 표면경화라 부른다. 이와 같은 표면경화 후기의 목재를 거단하면 내측으로 굽는다. 이러한 응력을 제거하기 위해 건조 종료 후 컨디셔닝(conditioning)처리를 한다. 표면경화는 할렬과 틀어짐을 유발하며 고함수율 또는 고비중의 목재에서 잘 나타난다. 할렬은 표면경화의 증거이고, 표면경화는 증기처리에 의해 완화해 주지 않으면 내부할렬을 유도한다.

표면경화가 발생한 목재는 아주 얇은 두께의 판재로 제재하는 경우를 제외하면 사용상 문제가 되지는 않는다. 또한 일반적인 건축부재로 사용되는 경우에는 거의 문제가 되지 않는다. 표면경화가 발생한 목재는 필요한 경우 건조가 끝나는 시점에서 짧은 시간동안 높은 온도와 습도조건을 적용하여 응력을 제거할 수 있다.

건조응력을 측정하는 방법에는 슬라이스법(slice test), 프롱법(prong test), 분할법(strip

test) 등이 있으며 표면경화의 정도는 표면경화율로 나타낸다. 건조응력을 측정하는 방법 중에서 프롱법에 의한 표면경화율이 0이면 없음, 0~33%이면 경미(slight), 33~67%이면 보통(medium), 67~100%이면 심함(severe), 0% 이하이면 역표면경화라고 표현한다. 그리고 표면경화율은 다음 식으로 구할 수 있으며 프롱테스트 후에 표면경화의 성질과 거동은 그림 25와 같다.

$$\text{표면경화율(\%)} = (1 - B/A) \times 100$$

본 실험에서는 프롱법을 채택하여 건조응력을 측정하였으며 실험에 이용된 시편들이 건조용 부재로 사용되는 규격의 목재를 이용하였기 때문에 일반열기건조스케줄을 적용한 두께 25mm와 50mm의 국산 소나무재만을 프롱법으로 측정하여 표면경화율의 정도를 알아보고자 하였다.

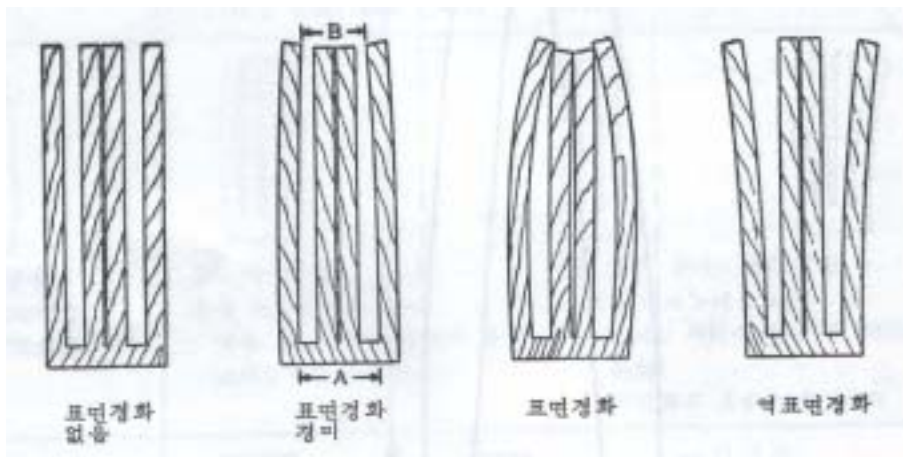


그림 25. 표면경화율 측정방법.

제 4 절 건조 소요에너지

적산 전력계를 사용하여 건조에 소요되는 전기에너지를 측정하여 증발수분 kg 당 소요되는 전기에너지(kWh)를 산출하였다. 그리고 전자동 소형 목재건조 시스템의 열에너지 효율을 측정하기 위하여 4가지의 기본적인 열요구인자들을 계산하였다. 열요구인자의 계산은 다음과 같다.

가) 木質의 온도상승을 위한 열에너지

모든 물질은 열을 흡수할 수 있는 능력이 있다. 온도상승을 위해 흡수해야만 될 열에너지의 양은 가열전과 후의 온도차, 가열할 물질의 양 및 그 물질의 열용량(thermal capacity) 또는 비열(specific heat) 등에 따라 변화한다. 물질의 열용량은 실험적으로 구해지며, 일정 온도에서는 변함이 없다.

Skaar(1972)에 따르면 목질 또는 전건목재의 열용량은 식(1.1)과 같으며, 일정 온도범위에서는 식(1.2)와 같이 표현된다.

$$C_w = 0.268 + 0.0011T \text{ ----- (1.1)}$$

$$= 0.268 + 0.00055(T_1 + T_0) \text{ ----- (1.2)}$$

C_w : 목질의 열용량(cal/g°C)

T_1 : 목질의 가열전 초기온도(°C)

T_0 : 목질의 가열후 온도(°C)

실제 현장에서 T_1 은 야적장에서의 목재의 온도 또는 건조실 외부의 온도가 될 것이며 T_0 은 건조과정중 최고온도가 될 것이다. 목재의 전건중량은 잔적된 목재의 총 체적에 그 수종의 비중을 곱하면 얻을 수 있다. 따라서 목질의 온도상승을 위한 열에너지는 다음의 식(1.3)으로 구할 수 있게 된다.

$$H_1 = 1000 \times ODW \times C_w \times (T_0 - T_1) \text{ ----- (1.3)}$$

H_1 : 목질의 가열에 소요되는 열에너지(cal)

ODW : 잔적전체의 전건중량(kg)

나) 흡습력(hygroscopic force) 극복을 위한 열에너지

목재는 수증기가 쉽게 부착되어 비교적 약한 결합으로 유지될 수 있는 화학적 구성을 하고 있다. 목재가 거의 전건 상태인 경우에는 그 결합은 상대적으로 강인해지지만 섬유포 화점 이상에서는 자유수의 흡습력은 없다. 1파운드의 목질이 생채상태에서 전건 될 때까지 흡습력을 극복하기 위하여 소요되는 열에너지는 평균 34Btu(18.889 cal/g)임이 실험을 통해 밝혀졌다. 이 열에너지는 탈착열(heat of desorption)로 알려져 있다.

탈착열은 함수율에 따라 달라지므로 Skaar(1972)가 제시한 식(1.4)를 이용하여 구한 후 잔적의 전건무게를 곱하여 흡습력 극복을 위한 열에너지를 구하였다.

$$H_{de} = 10^{1.2335-5.408Me} \text{ ----- (1.4)}$$

$$H_2 = 1000 \times ODW \times H_{de} \text{ ----- (1.5)}$$

H_{de} : 목질의 단위무게당 탈착열(cal/g)

Me : 목표(최종) 함수율(분율)

H_2 : 흡습력을 극복하기 위한 열에너지(cal)

다) 목재내 잔존수분의 가열을 위한 열에너지

열기건조란 목재를 전건상태까지 건조하는 것은 아니므로 항상 목재내에는 약간의 수분이 남아있게 마련이다. 건조과정중에 건조실 온도는 상승되기 마련이므로 잔존수분의 온도상승에도 약간의 열에너지가 소모되며, 그 열량은 목질가열의 경우와 같은 방법으로 구하면 된다. 물의 열용량은 온도에 따라 변화하나 대개의 경우 1cal/g℃를 적용한다.

$$H_3 = 1000 \times ODW \times M_R \times 1 \times (T_0-T_1) \text{ ----- (1.6)}$$

H_3 : 목재내 잔존수분의 가열을 위한 열에너지(cal)

M_R : 잔존수분의 양(분율)

라) 목재로부터 제거될 수분의 가열과 증발을 위한 열에너지

목재로부터 제거될 수분에 의해 흡수된 열도 H_3 의 경우와 같은 방법으로 구할 수 있다. 그러나 공기순환에 의해 이 수분이 목재로부터 분리되기 위해서는 액체상태에서 기체상태로 상변환을 해야 하며, 이와 같은 상변환에는 주어진 온도에 따라 상당한 열에너지가 요구된다. 따라서 이 열요구인자는 건조스케줄 각 단계의 온도수준마다 계산해야 하며, 각

단계별 열량을 모두 종합하면 요구되는 총 열에너지를 얻을 수 있다.

$$H_4 = \sum H_{4i}$$
$$= ODW \times 1000 \sum [\Delta mc_i \times (\Delta T_i \times 1 + h_{fgi})] \text{ ----- (1.7)}$$

H4 : 목재로부터 제거될 수분의 가열과 증발을 위한 열에너지(cal)

Δmc_i : 건조스케줄의 각 단계별 감소되는 함수율(분율)

ΔT_i : 각 단계별 건구온도와 외기 온도와의 차(°C)

1 : 물의 비열

제 5 장

전자동 제재목 소형 건조시스템의 성능

제 1 절 일반 열기건조스케줄 적용 건조실험

일반적으로 이용되고 있는 미국 임산물연구소에서 개발된 스케줄 중에서 T8F4의 건조스케줄을 적용하여 25mm, 50mm 두께의 국산 소나무 시편을 건조한 결과는 그림 26 및 그림 27과 같았다.

건조 후 목재의 목표함수율을 10%로 기준할 때 국산 소나무의 건조시간은 두께 25mm인 경우는 총 64시간이 소요되었고, 두께 50mm인 경우에는 총 233시간이 소요되었다.

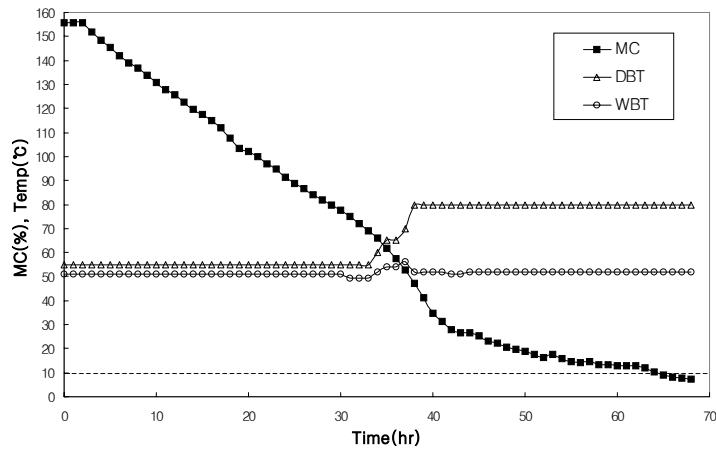


그림 26. 두께 25mm 국산 소나무의 일반 함수율 기준 스케줄 건조곡선.

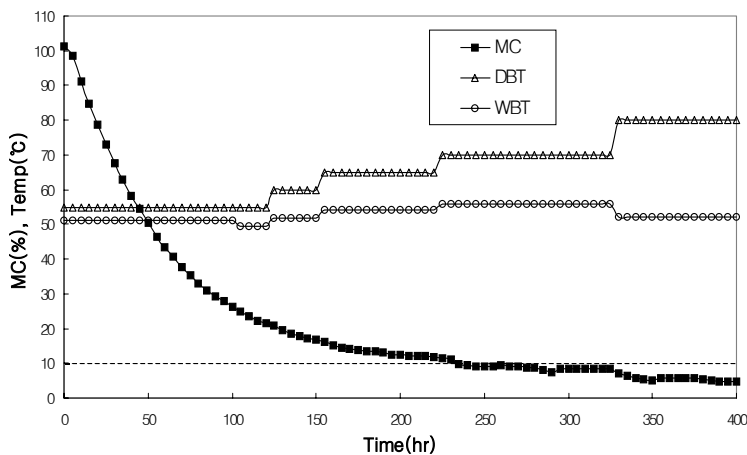


그림 27. 두께 50mm 국산 소나무의 일반 함수율 기준 스케줄 건조곡선.

제 2 절 시간 기준 건조스케줄 적용 건조실험

두께 50mm의 국산 소나무재에 시간 기준 건조스케줄을 적용하여 건조 실험을 한 결과 첫 번째 실험에서는 초기함수율 50.2%에서 목표함수율인 10%까지 걸린 시간은 61시간이었으며, 두 번째 실험에서는 초기함수율 63.8%에서 목표함수율인 10%까지 소요된 시간은 총 74시간이었다.

실험 결과는 그림 28 및 그림 29에 나타나 있다.

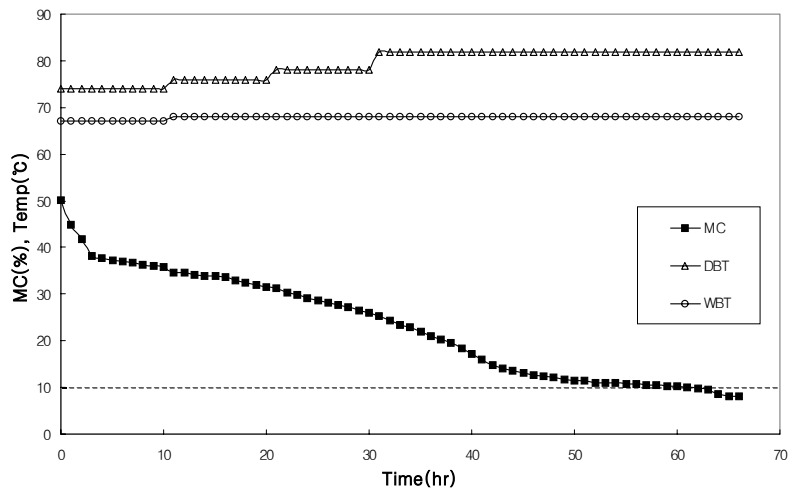


그림 28. 두께 50mm 국산 소나무의 시간기준스케줄 건조곡선(Test 1).

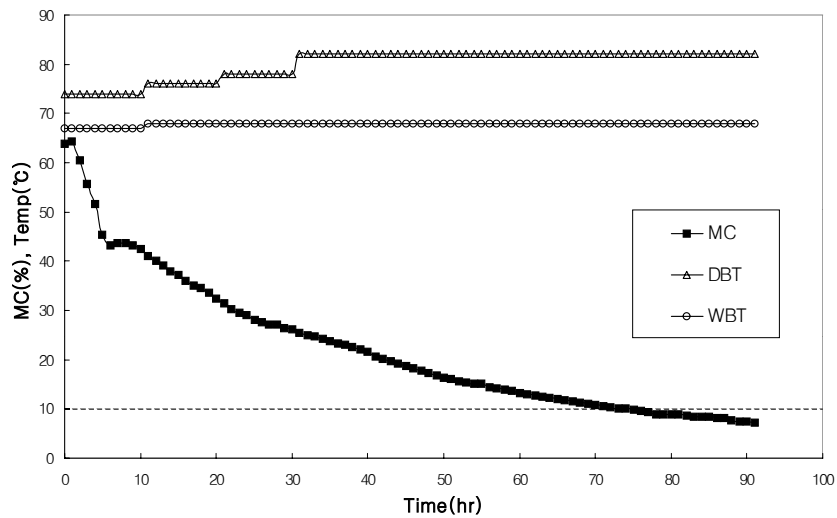


그림 29. 두께 50mm 국산 소나무의 시간기준스케줄 건조곡선(Test 2)

제 3 절 연속 온도상승 건조스케줄 적용 건조실험

국산 소나무재의 건조에 연속 온도상승 건조스케줄을 적용하였을 때 두께 25mm와 50mm의 건조실험결과는 그림 30 및 그림 31과 같았다.

시편의 목표함수율인 10%에 도달하기까지 소요된 건조시간은 두께 25mm인 시편의 경우는 약 35시간이 걸렸고, 두께 50mm인 시편은 목표함수율인 10%까지 65시간이 소요되었다.

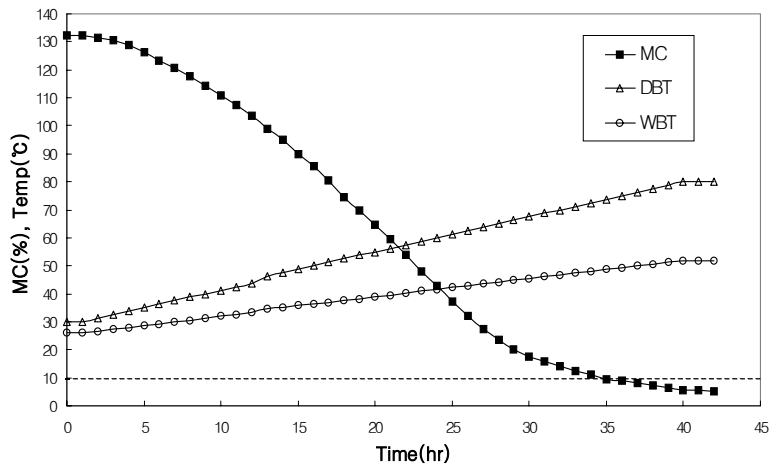


그림 30. 두께 25mm 국산 소나무의 연속온도상승스케줄 건조곡선.

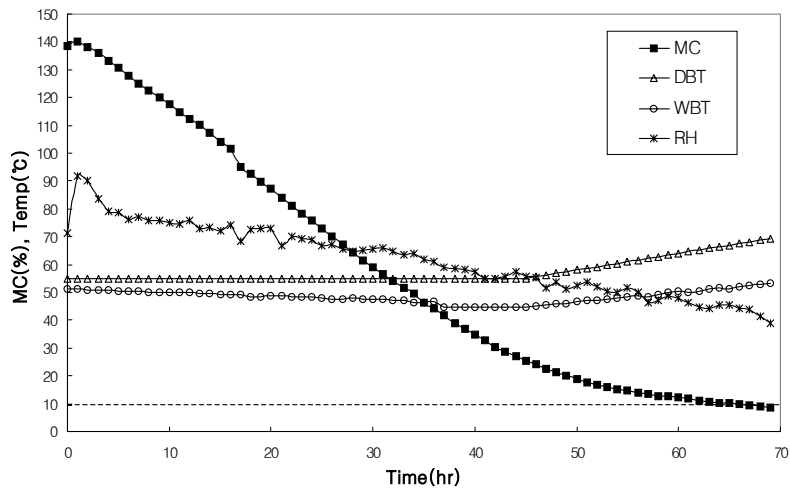


그림 31. 두께 50mm 국산 소나무의 연속온도상승스케줄 건조곡선.

제 4 절 습구온도조절 건조스케줄 적용 건조실험

습구온도조절 건조스케줄 적용 실험에는 우리의 전통가옥인 한옥이나 문화재 복원용으로 주로 사용되는 마루판 소재를 이용하였으며, 그 규격은 50×270×1800mm이었다. 모두 2회에 걸쳐서 실행하였으며, 건조 완료 후에는 목재의 건조결함과 건조효율 등 에너지 비용을 계산하였다.

습구온도조절 건조스케줄 적용 실험 결과 첫 번째 실험(Test 1)은 초기함수율 169.8%에서 시작하여 최종함수율 12.1%까지 총 123시간이 소요되었는데 이것은 건조 초기에 건조기 내부의 습도조절을 위해서 강제 배기를 하지 않았기 때문에 건조기 내부가 습기로 포화가 되어 건조가 지연되었다. 반면에 마루판의 품질은 대단히 양호하였다.

Test 1의 결과를 토대로 하여 두 번째 실험(Test 2)은 건조 중에 강제 배기의 조절을 통하여 건조시간을 조절하였으며 건조가 완료된 후에 적산전력계를 통하여 건조 효율을 측정하였다. Test 2의 건조 결과는 초기함수율 150.4%에서 최종함수율 11.5%까지 총 110시간이 소요되었으며, 건조품질도 양호하였다.

습구온도조절 건조스케줄을 적용하여 마루판을 건조한 실험결과는 그림 32 및 그림 33과 같았다.

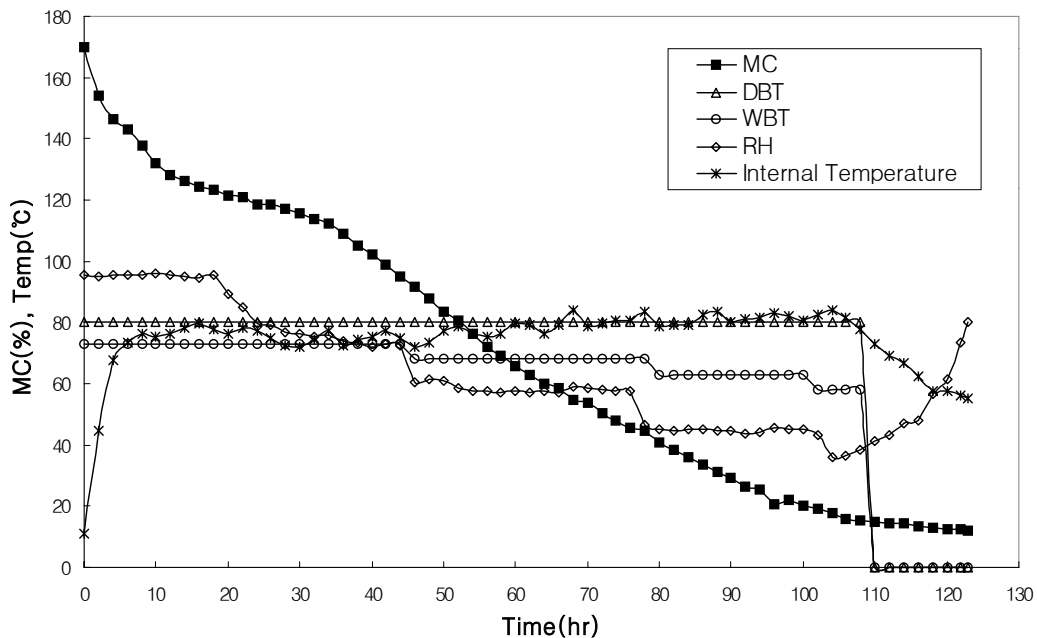


그림 32. 규격 50×270×1800mm 국산 소나무의 습구온도조절스케줄 건조곡선(Test 1).

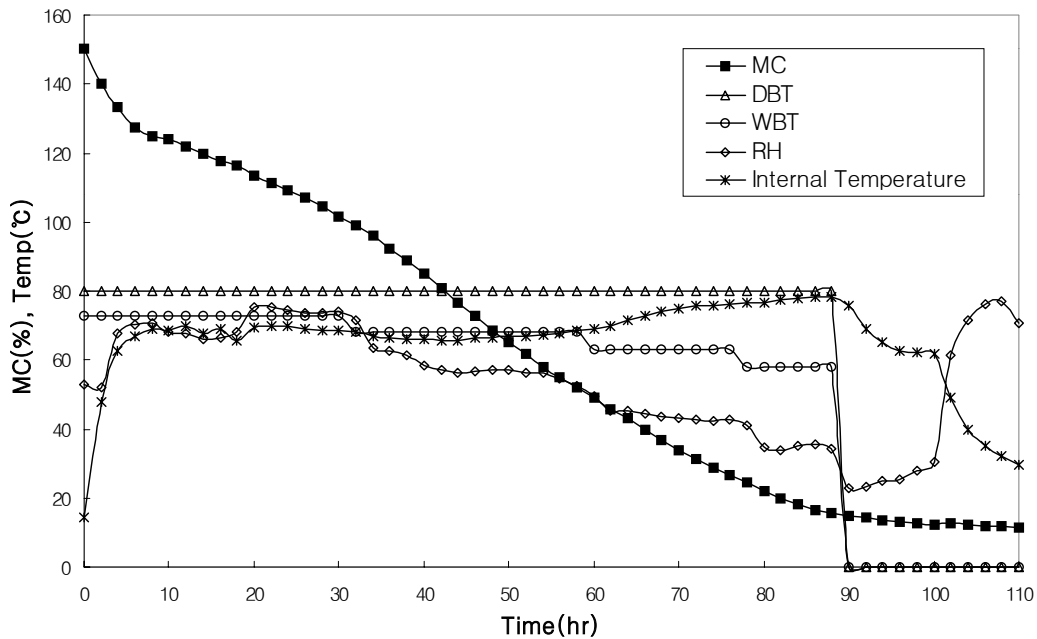


그림 33. 규격 50×270×1800mm 국산 소나무의 습구온도조절스케줄 건조곡선(Test 21).

제 5 절 건조속도

실험 결과 각각의 수종과 두께, 그리고 적용 스케줄에 따른 건조속도(drying rate)는 표 9와 같았다. 표 9에서 전체는 제재목의 생재함수율에서 15%까지의 건조속도를 의미한다.

국산 소나무 시편 중 두께 25mm의 건조실험 결과는 일반열기건조스케줄(CKDS)의 평균 건조속도가 2.55%/hr이었으며, 연속온도상승스케줄(CRT)은 3.78%/hr로서 연속온도상승스케줄의 건조속도가 상당히 빨랐으며 건조속도 향상효과는 약 48%에 달하였다. 연속온도상승스케줄 건조실험 중에 시편에 다수의 내부할렬 발생이 육안으로 관찰이 되었고 표면할렬은 없었다. 이 점으로 미루어 보아 구조재로서의 목재건조인 점을 감안할 때 커다란 변수가 되지는 않을 것으로 사료가 되지만, 건조품질의 개선을 위해서는 연속 온도상승 건조스케줄의 건조속도를 적절하게 조정하는 기술이 필요할 것으로 생각된다.

국산 소나무 두께 50mm 제재목의 경우는 일반열기건조스케줄의 평균건조속도는 0.53%/hr, 시간 기준 건조스케줄(TS) 적용실험에서는 각각 1.17, 0.9%/hr이었다. 그리고 연속온도상승스케줄 적용실험의 경우는 2.29%/hr이었고, 습구온도조절 건조스케줄(DRS)의 건조속도는 각각 1.4, 1.52%/hr이었다. 여러 가지 건조스케줄을 적용한 50mm 두께의 소나무

건조실험에서는 연속온도상승스케줄을 적용했을 때의 건조 속도가 2.29%/hr로 가장 빨랐으나, 건조 중 함수율 40~25%사이에서 내부할렬의 발생빈도가 높아 건조품질을 저하시켰다. 반면 습구온도조절 건조스케줄의 경우는 건조결합발생이 거의 없었다. 따라서 두께 50mm 국산 소나무의 적정한 건조속도와 양호한 건조품질을 위해서는 습구온도조절 건조스케줄을 적절히 적용하는 것이 바람직할 것으로 생각되며, 건조 중에 건조기 내부의 상대습도를 어떻게 제어하느냐에 따라서 건조속도와 건조품질을 상승시킬 수 있을 것이고, 결과적으로 경제적으로 많은 도움이 될 것으로 생각된다.

표 9. 건조스케줄에 따른 국산 소나무의 건조속도 비교

수종	건조 스케줄	두께 (mm)	건조속도(%/hr)					전체 (%/hr)
			생재-100	100-40	40-25	25-15	15-최종	
국 산 소 나무	CKDS	25	2.64	3.33	2.50	1.00	0.60	2.55
	CKDS	50	0.33	0.97	0.39	0.34	0.04	0.53
	TS	50	-	5.10	0.52	0.33	1.40	1.17
	TS	50	-	1.98	0.75	0.45	0.2	0.90
	CRT	25	2.47	5.45	3.75	3.33	0.9	3.78
	CRT	50	2.40	2.70	2.14	1.10	0.42	2.29
	DRS1	50	1.70	1.54	1.00	0.67	0.22	1.40
	DRS2	50	1.60	1.70	1.36	0.83	0.17	1.52

제 6 절 건조결합

6.1. 틀어짐

건조결합은 먼저 상부하중을 가했을 때와 하지 않았을 때의 틀어짐(warping) 정도의 차이를 알아보려고 비중이 다른 소나무(KRP)와 낙엽송(L)을 시편으로 하여 실험을 하였다. 시편의 규격은 50×100×2400mm로 하였고, 상부하중을 가한 것과 하지 않은 것으로 나누고 똑 같은 건조조건으로 건조를 실행하여 건조가 종료된 후에 틀어짐(warping)의 치수를 측정하였다.

소나무와 낙엽송을 대상으로 한 고온건조실험결과 상부하중을 가한 것이 하지 않는 것에 비해 그 틀어짐의 정도가 낮아 소나무의 경우 23%, 그리고 낙엽송의 경우 42% 정도 틀어짐 수준을 낮출 수 있었다. 그 측정결과는 표 10과 그림 34와 같았다.

표 10. 처리방법별 두께 50mm 국산 소나무와 낙엽송의 틀어짐 정도

수종	틀어짐	평균		향상효과	
		상부하중 처리	무처리	(%)	평균
국 산 소나무	twist	7.4	7.7	4	23
	cup	0.8	1.1	27	
	crook	1.9	3.2	41	
	bow	1.8	2.3	22	
국 산 낙엽송	twist	3.8	6.8	44	42
	cup	0.5	0.9	44	
	crook	2.1	3.1	32	
	bow	2.2	4.1	46	

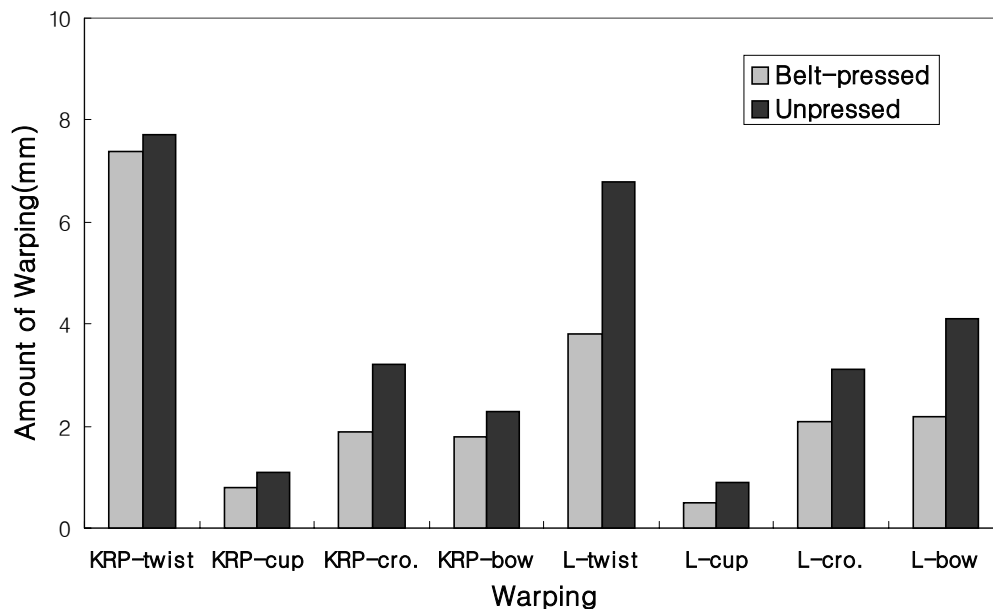


그림 34. 두께 50mm 국산 소나무와 낙엽송의 상부하중 처리에 의한 틀어짐 억제효과.

다음으로 마루판용 국산 소나무(50×270×1800)를 상부하중을 가한 상태로 건조가 완료된 후에 뒤틀림(warping)을 2회 측정된 결과 비틀림(twist)의 수치가 높게 나왔고 그 밖의 너비굽음(cup), 측면굽음(crook), 길이굽음(bow)은 수치가 낮았다. 비틀림의 수치가 높게 나온 것은 건조 시편의 폭이 넓은 것과 연관이 있는 것으로 추정되며, 소나무 시편의 건조 후 뒤틀림은 표 11과 그림 35에서 보여 주고 있다.

표 11. 마루판용 두께 50mm 국산 소나무의 틀어짐 정도

수종	틀어짐	Test1 (mm)	Test2 (mm)	평균 (mm)
국산 소나무 (50×270×1800)	twist	12.4	11.5	11.95
	cup	1.7	1.2	1.45
	crook	0.9	1.0	0.95
	bow	1.3	1.2	1.25

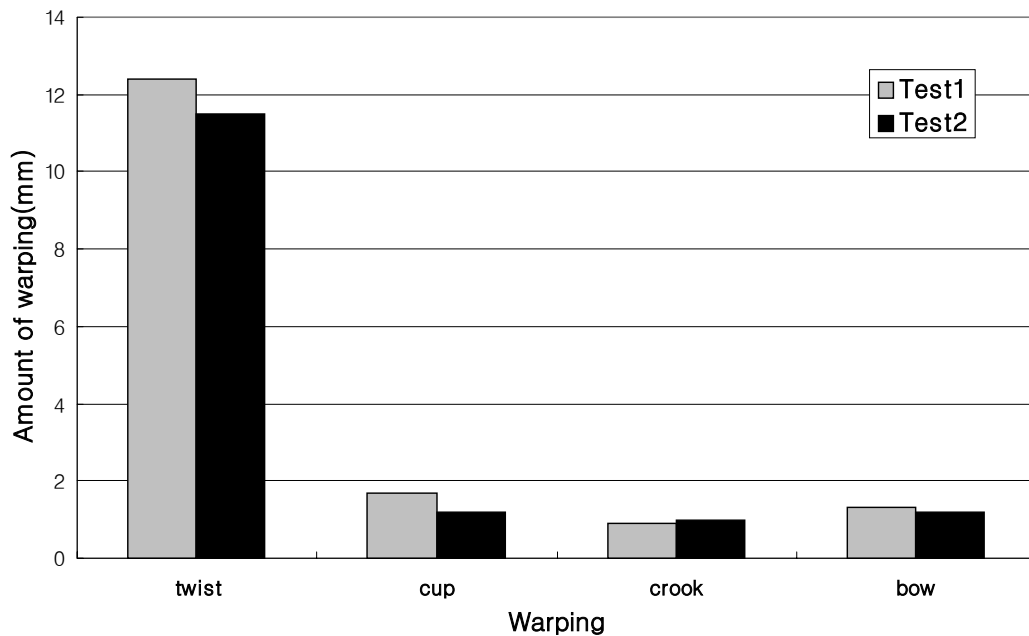


그림 35. 마루판용 국산 소나무의 틀어짐 비교.

6.2. 건조응력

1) 함수율 분포

건조응력을 측정하기에 앞서 일반열기건조스케줄을 적용한 두께 25mm와 50mm 소나무 시편의 건조 완료 후 목재 내부와 외부의 함수율 분포를 알아 보았다. 실험 결과 두께 25mm 소나무 시편의 내부함수율은 평균 7.7%이었고, 외부 함수율은 평균 6.8%로 나타났으

며, 내부 함수율이 외부 함수율에 비하여 0.9%가 더 높게 나타났다. 그리고 두께 50mm 소나무 시편의 내부함수율은 평균 5.0%이었고, 외부 함수율은 평균 3.8%로 나타났으며, 내부 함수율이 외부 함수율에 비하여 1.2%가 더 높게 나타났다. 두께에 따른 함수율 분포에 대한 실험결과는 표 12와 같았다.

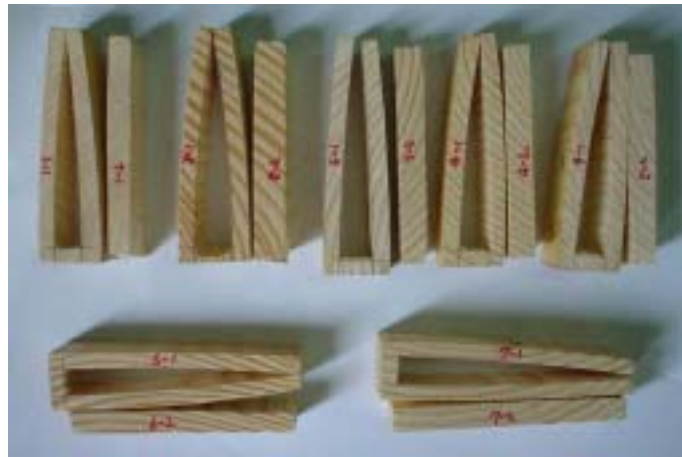


그림 36. 두께 25mm와 50mm 국산 소나무의 건조응력 측정을 위한 prong-test 후 모습.

표 12. 두께 25mm와 50mm 국산 소나무의 건조후 함수율 분포

두께 (mm)	평균 함수율 (%)		함수율 경사(%) (내부-외부)
	내부	외부	
25	7.7	6.8	0.9
50	5.0	3.8	1.2

2) 표면경화율

프롱법을 이용하여 건조후 두께 25mm와 50mm 소나무 시편의 표면경화율을 측정 한 결과 두께 25mm 소나무 시편의 경우는 평균 표면경화율이 91%로서 그 정도가 심하였고, 두께 50mm 소나무 시편의 경우는 평균 표면경화율이 12%로서 표면경화의 정도가 경미하였다. 본 실험의 결과는 표 13과 같았다.

표 13. 두께 25mm와 50mm 국산 소나무의 표면경화율

두께 (mm)	표면경화율 (%)
25	91
50	12

6.3. 수지고정

침엽수류의 건조에서 건조후 가장 문제점으로 대두되고 있는 것 중의 하나가 목재내 함유되어 있던 수지가 고정되지 못하고 건조후 사용중 용출되는 문제이다. 사용중 수지가 용출되면 사용자의 의복 등 묻어 곤란한 경우가 빈번히 발생하게 된다. 이와 같은 문제점이 발생하는 가장 큰 원인은 낮은 온도에서 불충분하게 건조하였거나 고온이라 할지라도 충분한 체류시간을 주지 않았기 때문이다. 그림 37은 본 연구에서 건조된 국산 소나무의 건조후 모습으로 목재내 수지가 건조중 대부분 용출되어 굳은 모습으로 이처럼 완벽히 건조된 목재는 차후 사용 중에 수지가 용출되어 야기되는 문제점이 없어 충분한 경쟁력이 될 것으로 판단된다.

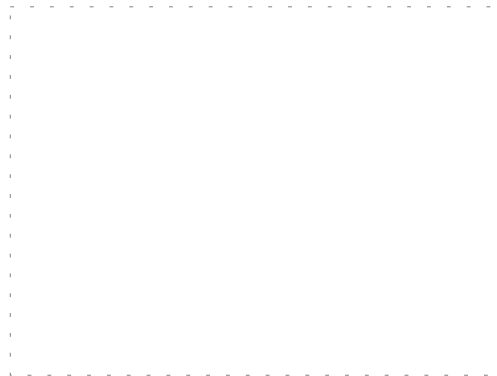


그림 37. 건조후 수지가 고정된 모습.

제 7 절 열효율

본 실험에서는 건조 중에 소비되는 총에너지의 산출을 위하여 국산 소나무 마루판용 제재목을 이용하였다. 국산 소나무 마루판용 제재목의 규격은 50×270×1800mm(0.0243m³)이며, 1회 건조 시에 39개의 제재목이 건조되어지므로 총 용적은 약 0.9477m³이 된다. 그리고 소나무 마루판의 건조 전 총 중량은 815.46kg이었으며 건조완료 후의 중량은 524.34kg이었다. 따라서 건조가 완료되었을 때는 총 291.12kg의 목재 내부의 수분이 증발되었음을 알 수 있다. 소나무 마루판의 건조 중에 소비된 총 전력량은 450kWh(약 386,930kcal)이었다. 결국, 수분 1kg을 제거하기 위하여 약 1.55kWh(약 1,329kcal/kg)의 전력이 소비되었음을 알 수 있다.

건조 후 최종함수율(FMC)이 11.5%이었으므로 추정 전건무게는 약 470.26kg이 되며, 따라서 초기함수율(IMC)은 73.4%가 된다.

80℃에서 물의 증발열은 약 551.57kcal/kg이므로 총증발열은 약 160,573kcal가 되며, 따라서 본 실험에 이용된 산업용 전자동 소형 목재건조기의 열효율은 약 41.5%가 된다.

다음으로 열요구인자를 구하여 보면 다음과 같다.

가. 목질의 온도상승을 위한 열에너지

$$\begin{aligned} \text{목질의 열용량}(C_w) &= 0.268 + 0.0011T \\ &= 0.268 + 0.00055(T_1 + T_0) \\ &= 0.3199 \text{ (cal/g}^\circ\text{C)} \end{aligned}$$

$$\begin{aligned} \text{목질의 온도상승을 위한 열에너지}(H_1) &= 1000 \times \text{ODW} \times C_w \times (T_0 - T_1) \\ &= 9,853,569 \text{ (cal)} = 9,854 \text{ (kcal)} \end{aligned}$$

나. 흡습력(hygroscopic force) 극복을 위한 열에너지

$$\begin{aligned} \text{목질의 단위무게당 탈착열}(H_{de}) &= 10^{1.2335-5.408Me} \\ &= 4.09 \text{ (cal/g)} \end{aligned}$$

$$\begin{aligned} \text{흡습력을 극복하기 위한 열에너지}(H_2) &= 1000 \times \text{ODW} \times H_{de} \\ &= 1,923,363 \text{ (cal)} = 1,923 \text{ (kcal)} \end{aligned}$$

다. 목재내 잔존수분의 가열을 위한 열에너지

$$\begin{aligned} & \text{목재내 잔존수분의 가열을 위한 열에너지}(H_3) \\ & = 1000 \times \text{ODW} \times M_R \times 1 \times (T_0 - T_1) \\ & = 3,542,233 \text{ (cal)} = 3,542 \text{ (kcal)} \end{aligned}$$

라. 목재로부터 제거될 수분의 가열과 증발을 위한 열에너지

$$\begin{aligned} & \text{목재로부터 제거될 수분의 가열과 증발을 위한 열에너지}(H_4) \\ & = H_{4i} \\ & = \text{ODW} \times 1000 \times [\Delta mc_i \times (\Delta T_i \times 1 + h_{f_{gi}})] \\ & = 176,625,249 \text{ (cal)} = 176,625 \text{ (kcal)} \end{aligned}$$

만일 상기의 열요구인자들을 모두 건조에 필수적인 에너지로 가정한다면 그 합은 191,944kcal가 되어 전체적인 열효율은 약 49.6%가 된다.

제 8 절 건조비용 추정

여러 번의 건조실험 결과를 기준으로 하여 건조비용을 추정하므로써 본 건조시스템의 우수성을 확인하고자 하였다. 실험결과 두 가지 건조기 모두 건조초기에는 건조기 내부 및 목재의 온도를 정해진 건조온도까지 가열하기 위한 에너지가 다량 소요되었으나 이후 일정 온도 수준이 유지되면서 에너지 소비량이 급격히 감소되었다. 이는 수분증발과 건조실 벽체를 통한 열손실 및 기타 열손실의 보전에만 에너지가 소요되기 때문으로 판단되며, 건조기간을 통틀어 실제 해당 건조기의 총 용량의 30% 정도가 소요된 것으로 추정되었다.

표 14는 두께 50mm 국산 소나무를 기준으로 산업용 전기료를 1kWh당 60원으로 가정하여 계산한 건조비용으로 용량이 클수록 에너지 소비에 의한 건조비용이 감소됨을 알 수 있었다. 한편, 현재 인천시에서 운영되고 있는 건조대행업체의 두께 50mm 소나무의 건조대행비용이 1才(1/300m³)당 200원 수준임을 감안할 때 인건비와 감가상각비를 모두 고려하여도 매우 저렴하게 건조를 수행할 수 있어 본 연구에 의해 개발된 목재 건조시스템의 경제성을 확인할 수 있었다.

표 14. 두께 50mm 국산 소나무 기준 소요 에너지 비용 추정

구 분	일반형 건조시스템	고온형 건조시스템
잔적실 폭(mm)	1,800	2,100
잔적실 높이(mm)	1,800	2,000
잔적실 길이(mm)	4,200	7,200
잔적 가능 목재 층수	26	29
최대 잔적 가능 목재용적(m ³)	8.0	20.0
건조소요일수(일)	4	4
총 전기용량(kW)	45	75
총 소요 전기비용(원)	69,120	115,200
목재 1才당 소요비용(원)	23.7	17.8

제 9 절 목조건축 및 문화재용 목재의 건조

본 연구를 통하여 개발된 건조기의 용도개발과 확장을 위하여 최근 관심이 모아지고 있는 목조건축용 및 목조 문화재용 목재의 건조를 실시하여 그 결과를 분석해 보았다. 그림 38은 두께 50mm와 두께 100mm의 목조건축용 국산 소나무로 두 두께를 동시에 투입한 장면으로 상부에 상부하중이 설치된 모습을 볼 수 있다.

한편, 그림 39는 목조 문화재 보수용 목재로 여러 가지 다양한 규격의 목재를 함께 투입하여 건조하였으며 만족스러운 결과를 얻을 수 있었다.

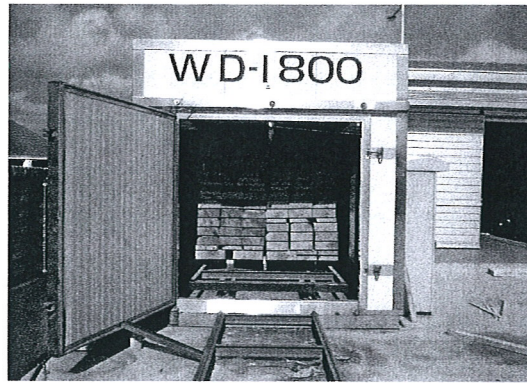


그림 38. 목조 건축용 목재의 잔적 장면.



그림 39. 목조 문화재 보수용 목재의 잔적 장면.

목조건축용 두께 50mm와 100mm 국산 소나무의 건조결과 얻은 건조곡선은 그림 40과 같았으며, 그림에서 보는 바와 같이 건조초기 및 중기에는 두께별 건조속도의 차이가 비교적 컸으나 건조가 진행되면서 두께별 건조속도 차이가 감소되어 상당히 만족스러운 결과를 얻을 수 있었다. 따라서 국산 소나무와 같이 비교적 건조가 용이한 수종의 경우에는 약간의 두께 차이는 무시하여도 무방할 것으로 사료된다.

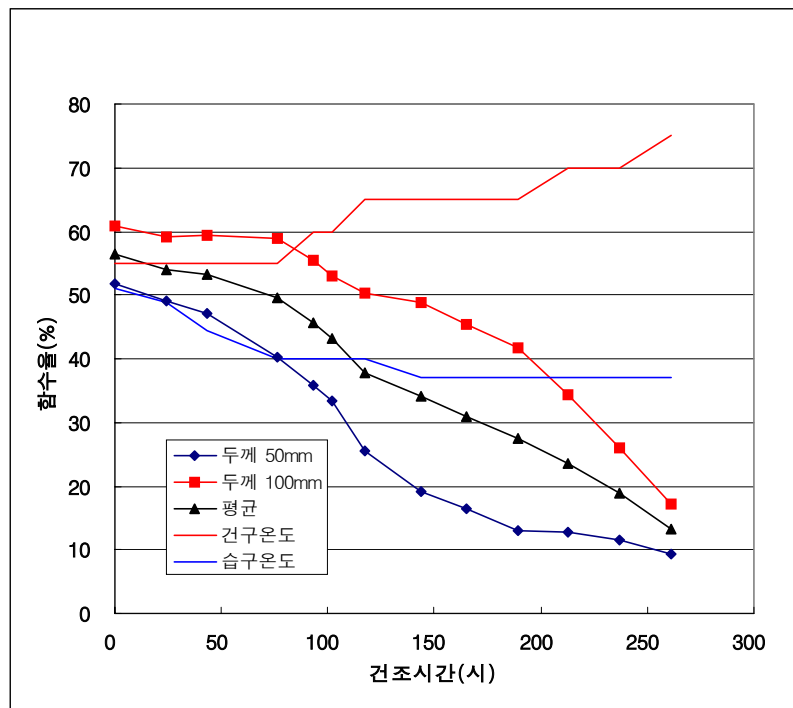


그림 40. 목조 건축용 국산 소나무의 건조곡선.

參考文獻

1. Wengert, E. M. and P. G. Evans. 1971. Automatic programming and control for steam-heated lumber dry kilns. *Forest Products Journal* 21(2): 56~59.
2. Holmes, S. M. and D. G. Arganbright. 1976. Drying tests with a weight-actuated lumber drying control system. *Forest Products Journal* 26(11): 31~39.
3. Little, R. L., W. W. Moschler, and R. L. Toennisson. 1986. Design of a computer based controller-recorder for an experimental dry kiln. *Forest Products Journal* 36(11/12): 72~74.
4. Shida, S., M. Chiba, and N. Nara. 1986. A computer-controlled system for operating kiln drying I. Conditions of practical system and an outline of a prototype one. *Hokkaido Forest Res. Ins.* 415(8): 1~11.
5. Shida, S. and N. Nara. 1988. A computer-controlled system for operating kiln drying II. How to solve the problem of output drifts of load cells caused by temperature changes. *Hokkaido Forest Res. Ins.* 2(1): 1~10.
6. Shida, S. and N. Nara. 1988. A computer-controlled system for operating kiln drying III. Performance of the "full-auto system" controlled by the continuous changing drying schedules on the basis of a moisture content measured by a moisture sensor consisted of a load cell. *Hokkaido Forest Res. Ins.* 2(2): 10~20.
7. Kunn, C. P. and H. F. Vermaas. 1988. The supervisory pc in process control systems with special reference to the kiln drying of wood. *Holz als Roh-und Werkstoff* 46: 27~31.
8. Kunn, C. P. and H. F. Vermaas. 1989. Computerized data-acquisition and control system for an experimental pressure steam dryer for lumber. *Holz als Roh-und Werkstoff* 47: 261~265.

9. Dedrick, D. S. 1968. Kiln drying process. U. S. Patent 3,404,464.
10. Microprocessor dry kiln control(Lignomat MP8). 1983. System optimizes kiln monitoring. Wood & Wood Products Sep.
11. Nassif, N. M. 1979. An approach to the drying of Eucalyptus Laevopinea and Eucalyptus Agglomerata using CRT. Forestry Commission of N. S. W.
12. Nassif, N. M. 1983. Continuously Varying Schedule(CVS) - A new technique in wood drying. Wood Science and Technology 17: 139~144.
13. Lee, Sang Bong and Hee Suk Jung. 1989. Studies on predicting the kiln drying time and moisture content of board and dimension lumber of Pinus densiflora using an internal moisture diffusion model of softwood. Korean Wood Sci. Tech. 17(3): 67~81.
14. Salamon, M. and S. McIntyre. 1970. Combination schedule improves drying of Western white spruce lumber. Forest Products Journal 20(7): 41~41.
15. Wengert. E. M.. 1974. Maximum initial moisture contents for kiln-drying 4/4 hardwood at high temperature. Forest Products Journal 24(8): 54~561.
16. 김우승, 이창식. 1995. 공업열역학. 東明社. 754.

제 6 장

종 합

본 연구에서는 국내 목재산업의 실정에 적합한 소형의 전자동 제재목 건조시스템을 개발하고자 하였다. 본 연구를 통하여 목재 잔적공간 규격 1800x1800x4200mm(13.6m³)인 일반형 건조시스템과 규격 2100x2000x7200mm(30.2m³)인 고온형 건조시스템을 설계, 제작하였다. 모든 제어와 계측은 user-interface를 위한 touch-panel형 컴퓨터시스템에 의하여 완전 자동화가 실현되었으며, 다양한 규격의 국내 침엽수 2수종(소나무, 낙엽송)에 대한 여러 가지 건조스케줄 적용실험을 통하여 건조속도, 함수율 분포, 건조응력, 건조결함, 소모에너지 등의 건조특성과 건조비용 등을 분석하므로써 본 건조시스템들의 성능을 평가하였다.

본 연구를 통하여 얻을 수 있었던 결론들은 다음과 같았다.

1. Touch-panel형 컴퓨터시스템에 현재 세계적으로 주로 사용되고 있는 수종들의 건조스케줄 database를 구축, 입력하므로써 목재건조 전문가가 아닌 운전자도 이용이 가능하였다. 한편, 실시간 data monitoring이 가능하였으며 모든 자료가 자동 저장되어 이후 기술축적에 근간이 될 것으로 기대된다.
2. 연속 온도상승 건조스케줄, 시간 기준 건조스케줄, 그리고 습구온도조절 건조스케줄의 적용실험을 하기에 앞서 두께 25mm와 50mm의 국산 소나무의 건조특성을 알아보기 위한 일반열기건조스케줄(미국 임산물연구소 스케줄) 적용실험 결과 두께 25mm 시편의 건조속도는 2.55%/hr로 건조속도가 빨랐고, 두께 50mm 시편의 건조속도는 0.53%/hr로서 그 속도가 느렸다. 일반 열기건조스케줄 적용실험 결과 시편의 내부할렬이나 표면할렬이 육안으로 관찰되지 않을 정도로 그 품질이 양호하였다.
3. 위의 일반열기건조스케줄의 적용 실험 결과를 바탕으로 함수율을 기준으로 제어하는 개념에서 벗어나서 경과시간에 따라 건구온도와 습구온도를 제어하는 개념의 시간 기준 건조스케줄을 구상하여 적용해 보았다. 적용 대상 시편은 두께 50mm의 국산 소나무였고 10시간 간격으로 건구와 습구온도를 제어하였다. 그 결과 건조속도는 평균 1.04%/hr로 나타났으며, 좀 더 빠른 건조속도를 위해서는 건구와 습구온도의 변화폭을 조정하는 시간을 단축시키면 될 것으로 생각된다.
4. 위의 일반열기건조스케줄과 시간 기준 건조스케줄의 적용결과를 기반으로 하여 건조속도를 보다 더 빠르게 목재를 건조할 수 있도록 하기 위하여 두께 25mm와 두께 50mm의 소나무 시편으로 연속 온도상승 건조스케줄을 적용하였다. 그 결과 두께 25mm 시편의 경우는 3.78%/hr의 아주 빠른 건조속도를 나타내었으며, 두께 50mm 시편의 경우

는 2.29%/hr로 일반열기건조스케줄과 시간 기준 건조스케줄의 건조속도보다 더 빨랐다. 그러나 건조 품질은 두께 25mm 시편과 50mm 시편에 표면할렬은 없었지만 다수의 내부할렬이 발생하여 건조품질을 저하시켰다. 건축 부재의 건조라는 측면에서는 연속 온도상승 건조스케줄의 실험 결과 발생된 내부할렬이 구조적으로 커다란 영향이 없을 것으로 생각되며, 따라서 건축부재의 건조는 본 실험에서 적용한 연속 온도상승 건조스케줄을 적절하게 수정하여 건조를 실시한다면 건조시간의 감소와 내부할렬의 감소로 경제적인 이점을 발생시킬 것으로 생각된다.

5. 마지막으로 실험실에서 소량의 시편으로 건조특성의 패턴을 알기위하여 행해진 일반열기 건조스케줄과 연속 온도상승 건조스케줄의 결과를 토대로 하여 목재건조산업 현장에서와 비슷한 정도의 건조 물량을 가지고 산업적인 규모의 실험을 실행하였다. 이 실험에서는 두께 50mm, 폭 270mm의 규격으로 우리의 전통가옥의 마루판 용도로 사용되는 국산 소나무를 시편으로 하여 2회에 걸쳐 습구온도조절 건조스케줄을 적용하였다. 그 결과 건조속도가 평균 1.46%/hr로서 산업현장에서 적용할 만큼 충분히 빠른 속도였으며, 건조품질 또한 양호한 수준이었다. 따라서 두께 50mm 국산 소나무의 건조에는 습구온도조절 건조스케줄을 적용하는 것이 바람직 할 것으로 생각된다.
6. 함수율 분포는 건조응력의 측정을 위한 프롱테스트와 병행하여 이루어 졌으며, 건조가 완료된 후 건조된 목재의 내부함수율이 외부보다 더 높게 나타났으며, 시편의 두께가 두꺼워 질수록 그 차이가 커졌다. 표면경화율 측정결과 두께 25mm의 국산 소나무 시편은 91%로 경화율이 대단히 심하였고, 두께 50mm 시편의 경우는 12%로 그 정도가 경미하였다. 이와 같은 건조응력은 건조가 완료되는 시점에서 적당한 컨디셔닝(conditioning) 처리를 함으로써 응력을 감소시킬 수 있을 것이다.
7. 건조결함 중 틀어짐(warping)의 정도는 목재를 잔적한 후에 500kgf/cm²으로 상부하중을 가한 목재가 가하지 않은 목재에 비해서 국산 소나무의 경우는 약 23% 정도가 더 양호한 것으로 나타났고, 낙엽송의 경우는 약 42%가 더 양호한 것으로 나타났다. 두께 50mm 마루판용 국산 소나무 시편의 틀어짐은 비틀림(twist)이 평균 11.95mm로 높았고, 너비굽음(cup), 측면굽음(crook), 길이굽음(bow)가 각각 1.45, 0.95, 1.25mm로 낮았다. 비틀림의 수치가 높게 나온 것은 시편 폭(270mm)의 크기가 어느 정도 연관이 있는 것으로 생각된다.

8. 이번 실험을 통하여 산업용 전자동 소형목재 건조기의 소요에너지를 알아본 결과 소나무 마루판의 건조에 총 450kWh(약 386,930kcal)의 전력량이 소비되었으며, 목재 내 수분 1kg을 제거하는데 약 1.55kWh(약 1,329kcal/kg)의 전력량이 소모되었음을 알 수 있었다. 그리고 이와 같은 결과를 통하여 전자동 소형 목재건조기의 열효율은 약 41.5%로써 대단히 양호함을 알 수 있었다. 또한 Skaar(1972)가 제시한 목재건조에 있어서 4가지 열 요구인자들을 계산하여 전자동 소형 목재건조기의 전체적인 열효율을 계산해본 결과 49.6%라는 대단히 높은 수치를 나타내었다.

9. 본 연구에서 개발된 전자동 제재목 소형 건조시스템의 건조비용을 두께 50mm 국산 소나무를 대상으로 분석한 결과 일반형의 경우 에너지비용이 목재 1才당 23.7원, 고온형의 경우 목재 1才당 17.8원이 소요되어 충분한 경제력을 확보한 것으로 판단되며, 건조기의 용량이 클수록 소요비용이 절감되는 것으로 분석되었다.

부 록 1.

운영 프로그램

SOURCE FILE

```

//*****
// DEDryer.CPP
//*****
#include <vcl.h>
#include "SplashUnit.h"
#pragma hdrstop
USERES("DEDryer.res");
USEFORM("Main.cpp", Form1);
USEFORM("SchSet.cpp", Form2);
USEFORM("SplashUnit.cpp", SplashF);
USEFORM("Keyboard.cpp", VKeyF);

//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        SplashF = new TSplashF( Application );
        SplashF->Show();
        SplashF->Update();

        Application->CreateForm(__classid(TForm1), &Form1);
        Application->CreateForm(__classid(TForm2), &Form2);
        Application->CreateForm(__classid(TVKeyF), &VKeyF);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    return 0;
}
//-----

//*****
//Main.CPP
//*****
#include <vcl.h>
#pragma hdrstop

#include <stdio.h>
#include <string.h>
#include "Main.h"
#include "SchSet.h" // 스케줄 설정
#include "Math.h"
#include "Keyboard.h" //가상키보드
#include "infiles.hpp"

//#include <excpt.h>

//-----
#pragma package(smart_init)
#pragma link "IPSComm_OCX"
#pragma link "KnobDisp_OCX"

```

```

#pragma resource "*.dfm"

TForm1 *Form1;

struct DryData
{
    float DBV, WBV, wind, weight, SDBV, SWDBV;
    float pDBV, pWBV, pwind, pweight, pSDBV, pSWDBV;
};

struct Schedule
{
    int st, et;
    float DBT, WBT, wind, RH, EMC;
};

struct CRTS
{
    float dDBT0, dDBT1, mxDBT, stDBT;
    float dWBT0, dWBT1, mxWBT, stWBT;
};

Schedule Sche[40]; //건조스케줄
CRTS crtsch;      // CRT스케줄

int nSchedule;    //해당스케줄 갯수
int ScheduleType;
//DryData DD

float pval[7];
int count;        //토탈 저장데이터수

TDateTime StTime, StTime25; //시작시간, 25%까지 도달시간
int DSaveTerm;           //데이터 저장간격
String DSaveFname, HeadTitle; // 데이터 저장파일명, 건조설명
String TimeBuf;

float DBT, WBT, weight, MC, RH, wind, RMC; //실제 건조데이터
float SDBT, SWBT, Swind, IDBT;
float EMC;

float MMC, MTemp;

float MaxMC;
int SaveInt;
bool operationFLG, flag25; //작동중?
bool CtlFLG, counterflow; // 송풍방향
bool InLoop; // 제어루프에 있는가?
int counterflowtime; // 송풍역회전시간
unsigned long int ctrlcount;
String EditName;
TComponent *preComp;

int egcstop; // 비상중료되었는가?

```

```

float CSCALE[7]=(1, 1, 1, 1, 1, 1, 1); // 그래프의 Y축 스케일 성분
bool ErrorState; // 건조중 에러 발생 유무

float ITEMP[150]; // Set 데이터 저장스택
float DTEMP[150]; // Disp 데이터 저장스택
bool reachlast; // (MC스케줄)최종스케줄에 도달했는가?
bool limittime; // (시간스케줄) ..

void __fastcall FormCreate(TObject *Sender);
void __fastcall FormClose(TObject *Sender, TCloseAction &Action);

void __fastcall InitScheduleLoad(TObject *Sender);

void __fastcall GraphShow(TObject *Sender);
void __fastcall GraphOneDraw(TObject *Sender);
void __fastcall Delay(int Time);
bool __fastcall ScaleRead(TObject *Sender);

String __fastcall InputComp(TMSComm *RComp);
void __fastcall OutputComp(TMSComm *WComp, String buf);
float __fastcall ReadDBT(TObject *Sender, int nID, int nReg);
void __fastcall NX9Write(int addr, float d);
void __fastcall WindSetValue(TObject *Sender);

void __fastcall StartBtnClick(TObject *Sender);
void __fastcall BitBtn25Click(TObject *Sender);
void __fastcall BitBtn28Click(TObject *Sender);

void __fastcall Timer1Timer(TObject *Sender);
void __fastcall Timer3Timer(TObject *Sender);
void __fastcall Timer2Timer(TObject *Sender);
void __fastcall Edit9Exit(TObject *Sender);
void __fastcall BitBtn18Click(TObject *Sender);
void __fastcall BitBtn20Click(TObject *Sender);
void __fastcall BitBtn23Click(TObject *Sender);
void __fastcall SpeedButton1Click(TObject *Sender);
void __fastcall RestartBtnClick(TObject *Sender);
void __fastcall MChartClickSeries(TCustomChart *Sender, TChartSeries *Series, int ValueIndex,
    TMouseButton Button, TShiftState Shift, int X, int Y);
void __fastcall MChartMouseUp(TObject *Sender, TMouseButton Button, TShiftState Shift, int X, int Y);
void __fastcall Timer4Timer(TObject *Sender);
void __fastcall BitBtn2Click(TObject *Sender);
void __fastcall BitBtn3Click(TObject *Sender);
void __fastcall Edit14Enter(TObject *Sender);
void __fastcall BitBtn14Click(TObject *Sender);
void __fastcall BitBtn4Click(TObject *Sender);
void __fastcall BitBtn17Click(TObject *Sender);
void __fastcall IniSave(TObject *Sender);
void __fastcall IniLoad(TObject *Sender);
void __fastcall ScTableDrawCell(TObject *Sender, int ACol, int ARow, TRect &Rect, TGridDrawState State);
void __fastcall StepControl(TObject *Sender);

```

```

void __fastcall CRTControl(TObject *Sender);
void __fastcall CRTControl2(TObject *Sender);
void __fastcall Image1DbClick(TObject *Sender);
void __fastcall BitBtn6Click(TObject *Sender);
void __fastcall BitBtn7Click(TObject *Sender);

void __fastcall CheckMC(TObject *Sender);
void __fastcall WMCCBClick(TObject *Sender);
void __fastcall FormPaint(TObject *Sender);

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}

//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    String buf;

    DSaveFname = DSaveFnameE->Text;
    EditName = "";

    counterflow = false;

    MaxMC = -999;
    GraphShow(Sender);
    if(Comp->PortOpen != true) Comp->PortOpen = true;
    Comp->InBufferCount =0;
    Comp->InputLen =0;

    if(MComp->PortOpen != true) MComp->PortOpen = true;
    MComp->InBufferCount =0;
    MComp->InputLen =0;

    PageControl1->ActivePage = TabSheet2;
    InitScheduleLoad(Sender);

    IniLoad(Sender);

    switch(ScheduleType)
    {
        case 0 :
        case 1 :
        case 2 : SCTNB1->PageIndex = 0; break;
        case 3 : SCTNB1->PageIndex = 1; break;
        case 4 : SCTNB1->PageIndex = 2; break;
    }
}

//-----
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
    BitBtn25Click(Sender);
}

```

```

Delay(100);
IniSave(Sender);
if(Comp->PortOpen)
    Comp->PortOpen = false;
if(MComp->PortOpen)
    MComp->PortOpen = false;
}

//-----
void __fastcall TForm1::IniLoad(TObject *Sender)
{
    TIniFile *ini;
    ini = new TIniFile(ChangeFileExt( Application->ExeName, ".INI" ) );

    Edit2->Text = ini->ReadString( "WOOD_INFORM", "Species", "소나무" );
    InitWE->Text = ini->ReadString( "WOOD_INFORM", "Wg", "300" );
    OvenWE->Text = ini->ReadString( "WOOD_INFORM", "Wo", "200" );

    InitMCE->Text = ini->ReadString( "WOOD_INFORM", "MCo", "50");
    LastMCE->Text = ini->ReadString( "WOOD_INFORM", "MCt", "10");

    Edit5->Text = ini->ReadString( "WOOD_INFORM", "SizeT", "25" );
    Edit6->Text = ini->ReadString( "WOOD_INFORM", "SizeW", "100" );
    Edit7->Text = ini->ReadString( "WOOD_INFORM", "SizeL", "500" );

    RWgtE->Text = ini->ReadString( "WOOD_INFORM", "RWgt", "1000");
    LWgtE->Text = ini->ReadString( "WOOD_INFORM", "LWgt", "1000");

    HeadTitle = ini->ReadString( "Drying_INFORM", "Title", "전자동 열기 건조시스템");
    Caption = "전자동 열기 건조시스템 -" + HeadTitle;

    Edit14->Text = ini->ReadString( "CTL_ENV", "WindFRTerm", "1" );
    DSaveTE->Text = ini->ReadString( "CTL_ENV", "DataSavingTerm", "1" );
    // DCtrlTE->Text = ini->ReadString( "CTL_ENV", "CtrlTerm", "10");

    DSaveFname = ini->ReadString( "CTL_ENV", "SaveFName", "dry.dat" );
    DSaveFnameE->Text = DSaveFname;

    egcstop = ini->ReadInteger( "RUNSTOP", "EmcStop", 0 );
    StTime = ini->ReadDateTime( "RUNSTOP", "StartTime", 0 );

    ScheduleType = ini->ReadInteger( "SCHEDULE", "Type", 1 );
    nSchedule = ini->ReadInteger( "SCHEDULE", "Number", 10 );

    int nsc, i, j;
    String buf, buf2;

    for (i=0;i<nSchedule;i++)
    {
        if (ScTable->Cells[0][i] == "") break;

        buf = "";
        for(j=0;j<ScTable->ColCount;j++)
        {

```

```

        buf2 = ScTable->Cells[j][i];
        if (buf2 == "") buf2 = " ";
        buf = buf + buf2 + ",";
    }
    ini->ReadString( "SCHEDULE", IntToStr(i+1), buf);
}

dDBTSL->Caption = ini->ReadString( "CVT", "DS", "12.4");
dDBTEL->Caption = ini->ReadString( "CVT", "DM", "80");
dDBT0L->Caption = ini->ReadString( "CVT", "D0", "1.0");

dWBTS�->Caption = ini->ReadString( "CVT", "WS", "10.4");
dWBTEL->Caption = ini->ReadString( "CVT", "WM", "52");
dWBTOL->Caption = ini->ReadString( "CVT", "W0", "0.5");

WBTimeL->Caption = ini->ReadString( "CVT", "WT", "5");

crtsch.stDBT = StrToFloat(dDBTSL->Caption);
crtsch.mxDBT = StrToFloat(dDBTEL->Caption);
crtsch.dDBT0 = StrToFloat(dDBT0L->Caption);

crtsch.stWBT = StrToFloat(dWBTS�->Caption);
crtsch.mxWBT = StrToFloat(dWBTEL->Caption);
crtsch.dWBT0 = StrToFloat(dWBTOL->Caption);

delete ini;
}

//-----
void __fastcall TForm1::IniSave(TObject *Sender)
{
    TIniFile *ini;
    String buf;

    ini = new TIniFile(ChangeFileExt( Application->ExeName, ".INI" ) );
    ini->WriteString( "WOOD_INFORM", "Species", Edit2->Text );
    ini->WriteString( "WOOD_INFORM", "Wg", InitWE->Text );
    ini->WriteString( "WOOD_INFORM", "Wo", OvenWE->Text );

    ini->WriteString( "WOOD_INFORM", "MCo", InitMCE->Text);
    ini->WriteString( "WOOD_INFORM", "MCt", LastMCE->Text);

    ini->WriteString( "WOOD_INFORM", "SizeT", Edit5->Text );
    ini->WriteString( "WOOD_INFORM", "SizeW", Edit6->Text );
    ini->WriteString( "WOOD_INFORM", "SizeL", Edit7->Text );

    ini->WriteString( "WOOD_INFORM", "RWgt", RWgtE->Text );
    ini->WriteString( "WOOD_INFORM", "LWgt", LWgtE->Text );

    ini->WriteString( "Drying_INFORM", "Title", HeadTitle);

    ini->WriteString( "CTL_ENV", "WindFRTerm", Edit14->Text);
    ini->WriteString( "CTL_ENV", "DataSavingTerm", DSaveTE->Text);
    // ini->WriteString( "CTL_ENV", "CtrlTerm", DCtrlTE->Text);
}

```



```

ini->WriteString( "CTL_ENV", "SaveFName", DSaveFName );

ini->WriteInteger( "SCHEDULE", "Type", ScheduleType );
ini->WriteInteger( "SCHEDULE", "Number", nSchedule );

ini->WriteString( "CVT", "DS", dDBTSL->Caption);
ini->WriteString( "CVT", "DM", dDBTEL->Caption);
ini->WriteString( "CVT", "D0", dDBT0L->Caption);

ini->WriteString( "CVT", "WS", dWBTS�->Caption);
ini->WriteString( "CVT", "WM", dWBTEL->Caption);
ini->WriteString( "CVT", "W0", dWBTOЛ->Caption);
ini->WriteString( "CVT", "WT", WBTimeL->Caption);

ini->WriteInteger( "RUNSTOP", "EmcStop", 0 );

int nsc, i, j;
String buf2;

for (i=0;i<nSchedule;i++)
{
    if (ScTable->Cells[0][i] == "") break;

    buf = "";
    for(j=0;j<ScTable->ColCount;j++)
    {
        buf2 = ScTable->Cells[j][i];
        if (buf2 == "") buf2 = " ";
        buf = buf + buf2 + ",";
    }
    ini->WriteString( "SCHEDULE", IntToStr(i+1), buf);
}

delete ini;
}

//-----
float __fastcall HexToDig(String s)
{
    int i, j, mul;
    float val;

    val =0;
    for(i=1;i<=4;i++)
    {
        mul =1;
        for(j=1;j<=4-i;j++) mul = mul*16;

        if (s[i] >= 'A' && s[i] <= 'F') val += (s[i]-55)*mul;
        if (s[i] >= '0' && s[i] <= '9') val += (s[i]-48)*mul;
    }
    return val/10;
}

```

```

//-----
float __fastcall fEMC(float T, float RH)
{
    float H, TF, K1, K2, K3, tmpv, EMC;

    H = RH/100;
    TF = 9*T/5+32;
    tmpv = TF*TF;

    K1 = 3.73+0.3642*TF-0.001547*tmpv;
    K2 = 0.674+0.01053*TF-(1.714/10000)*tmpv;
    K3 = 216.9+0.1961*TF + 0.0572*tmpv;

    tmpv = K2*H;
    EMC = (K1*tmpv/(1+K2*tmpv)+tmpv/(1-tmpv))*1800/K3;

    return EMC;
}

//-----
float fWBD(float DBT, float RH)
{
    FILE *fp;

    float D1, D2, tp, L1, L2, L3, L4;
    float W1, W2, _tp, r1, r2;
    float WBD;
    int i, j, chkwp, nrow;
    char buf[120];
    char *p;

    if (RH >=100) return 0.0;

    fp = fopen("RH.TAB", "rt");
    if (fp==NULL)
    {
        Application->MessageBox("RH.TAB 파일을 찾을 수 없습니다.", "파일 열기 에러 !!", MB_OK);
        return 0;
    }
    fseek(fp, 0, SEEK_SET);

    nrow = (int)(DBT / 2);

    fgets(buf, 140, fp);
    for (i=0;i<nrow;i++)
        fgets(buf, 140, fp);

    p = strtok(buf, " ");
    D1 = atof(p);
    for (i=0;i<=31;i++)
    {
        p = strtok(NULL, " ");
        if (p)
        {
            _tp = atof(p);
            if (_tp <= RH)

```

```

        {
            L2 = _tp;
            break;
        }
        L1 = _tp;
    }
}

fgets(buf, 140, fp);
p = strtok(buf, " ");
D2 = atof(p);

for (i=0;i<=31;i++)
{
    p = strtok(NULL, " ");
    if (p)
    {
        _tp = atof(p);
        if (_tp <= RH)
        {
            L4 = _tp;
            chkwp = i;
            break;
        }
        L3 = _tp;
    }
}
fseek(fp, 0, SEEK_SET);
fgets(buf, 140, fp);
p = strtok(buf, " ");

for (i=1;i<chkwp;i++)
    p = strtok(NULL, " ");
W1 = atof(p);

p = strtok(NULL, " ");
W2 = atof(p);
fclose(fp);

r1 = L1+((DBT-D1) * (L3-L1) / (D2-D1));
r2 = L2+((DBT-D1) * (L4-L2) / (D2-D1));
if (r1==r2)
    WBD = r1;
else
    WBD = W1+((W2-W1) * (RH-r1) / (r2-r1));

return WBD;
}

//-----
float __fastcall fRH(float DBT, float WBD)
{
    FILE *fp;
    float D1, D2, tp, L1, L2, L3, L4;
    float W1, W2, _tp, r1, r2;
    float RH;
    int i, j, chkwp, nrow;

```

```

char buf[120];
char *p;
String Buf, msgb;

if (WBD <= 0) return 100.0;
if (WBD >40) return 0.0;

msgb = "의미없는 값이 들어왔습니다. 상대습도를 0%로 설정하겠습니다.";
fp = fopen("RH.TAB", "rt");
if (fp==NULL)
{
    Application->MessageBox("RH.TAB 파일을 찾을 수 없습니다.", "파일 열기 에러 !!", MB_OK);
    return 0.0;
}
fseek(fp, 0, SEEK_SET);
fgets(buf, 140, fp);
W2 = 40.0;
p = strtok(buf, " ");
for (i=0;i<=31;i++)
{
    p = strtok(NULL, " ");
    if (p)
    {
        _tp = atof(p);
        if (_tp > WBD)
        {
            W2 = _tp;
            break;
        }
        W1 = _tp;
    }
}
chkwp = i;

nrow = (int)(DBT / 2);

D1 = nrow * 2;
D2 = D1 + 2;

for (i=0;i<nrow;i++)
    fgets(buf, 140, fp);

p = strtok(buf, " ");
for (i=0;i<=chkwp;i++)
{
    p = strtok(NULL, " ");
    if (p == NULL) { Application->MessageBox(msgb.c_str(), "입력 에러!!", MB_OK); fclose(fp);return 0.0;
}
}
L1 = atof(p);

p = strtok(NULL, " ");
if (p == NULL) { Application->MessageBox(msgb.c_str(), "입력 에러!!", MB_OK); fclose(fp);return 0.0; }

L2 = atof(p);

```

```

if (DBT != 100)
{
    fgets(buf, 140, fp);

    p = strtok(buf, " ");
    if (p == NULL) { Application->MessageBox(msgb.c_str(), "입력 에러!!", MB_OK); fclose(fp);return
0.0; }

    for (i=0;i<=chkwp;i++)
        p = strtok(NULL, " ");
    if (p == NULL) { Application->MessageBox(msgb.c_str(), "입력 에러!!", MB_OK); fclose(fp);return
0.0; }

    L3 = atof(p);

    p = strtok(NULL, " ");
    if (p == NULL) { Application->MessageBox(msgb.c_str(), "입력 에러!!", MB_OK); fclose(fp);return
0.0; }

    L4 = atof(p);
}
else
{
    L3 = L1; L4 = L2;
}
fclose(fp);
if((L1==0)&&(L2==0)&&(L3==0)&&(L4==0)) return 0.0;

r1 = L1-((WBD-W1) * (L1-L2) / (W2-W1));
r2 = L3-((WBD-W1) * (L3-L4) / (W2-W1));
if (r1==r2)
    RH = r1;
else
    RH = r1+((DBT-D1) * (r2-r1) / (D2-D1));

return RH;
}

//-----
void __fastcall TForm1::InitScheduleLoad(TObject *Sender)
{
    int i, j, r;
    char buf[100];
    char *p;
    String buf2;
    FILE *fp;

    fp = fopen("init.stb", "rt");
    if (fp==NULL) return ;

    fseek(fp, 0, SEEK_SET);
    fgets(buf, 95, fp);
    ScheduleType = atoi(buf);
    i = 0;
    while (!feof(fp))
    {
        fgets(buf, 95, fp);
        if(feof(fp)) break;

        p = strtok(buf, ",");

```

```

if (p == NULL) break;

for(j=0;j<ScTable->ColCount;j++)
{
    ScTable->Cells[j][i] = p;
    p = strtok(NULL, ",");
}
i++;
}
fclose(fp);
nSchedule = i;

switch (ScheduleType)
{
    case 0 :
        Label74->Caption = "합 수 율";
        Label75->Caption = "( % )";
        break;
    case 1 :
        Label74->Caption = "건설시간";
        Label75->Caption = "(hour)";
        break;
    case 2 :
        Label74->Caption = "건설시간";
        Label75->Caption = "(min)";
        break;
}
for (i=0;i<nSchedule;i++)
{
    if((i==0) && (ScTable->Cells[0][i] == "生")) Sche[i].st = 300;
    else Sche[i].st = ScTable->Cells[0][i].ToInt();

    Sche[i].et = ScTable->Cells[1][i].ToInt();

    Sche[i].DBT = ScTable->Cells[2][i].ToDouble();
    Sche[i].WBT = ScTable->Cells[3][i].ToDouble();
    if ((ScTable->Cells[4][i] == "") ||(ScTable->Cells[4][i] == " "))
        ScTable->Cells[4][i] = "1.0";
    Sche[i].wind = ScTable->Cells[4][i].ToDouble();

    Sche[i].RH = ScTable->Cells[5][i].ToDouble();
    Sche[i].EMC = ScTable->Cells[6][i].ToDouble();
}
}

//-----
void __fastcall TForm1::Delay(int Time)
{
    long int BeginCount;

    BeginCount = GetTickCount(); //시작 시간

    while ((int)(GetTickCount()-BeginCount) < Time)
    {
        Application->ProcessMessages(); //멀티테스킹을 위해서..
    }
}

```

```

//-----
void __fastcall TForm1::GraphOneDraw(TObject *Sender)
{
    if (count > MChart->BottomAxis->Maximum)
        MChart->BottomAxis->Maximum = MChart->BottomAxis->Maximum+2;

    //    Label78->Caption = IntToStr(count);
    //    Label79->Caption = FloatToStr(MChart->BottomAxis->Maximum);

    Series1->Add(DBT, TimeBuf, clRed);
    Series2->Add(WBT, TimeBuf, clBlue);
    Series3->Add(RH, TimeBuf, clBlack);
    Series4->Add(MMC, TimeBuf, clGreen);
    Series5->Add(IDBT, TimeBuf, 0x00400040);
}

//-----
void __fastcall TForm1::GraphShow(TObject *Sender)
{
    MCKnob->Pointers->Item(1)->Value = 0;
    WindKnob->Pointers->Item(1)->Value = 0;

    DBTKnob->Pointers->Item(1)->Value = 10;
    DBTKnob->Pointers->Item(2)->Value = 20;

    WBTKnob->Pointers->Item(1)->Value = 10;
    WBTKnob->Pointers->Item(2)->Value = 20;
}

//-----
String __fastcall TForm1::InputComp(TMSSComm *RComp)
{
    String buf;
    if(!RComp->PortOpen) return "";
    if(operationFLG == false) return "";

    buf = RComp->Input.ChangeType(varString);
    return buf;
}

void __fastcall TForm1::OutputComp(TMSSComm *WComp, String buf)
{
    if(operationFLG == false) return ;

    if(WComp->PortOpen)
        WComp->Output = buf;
}

// 로드셀 데이터 읽기-----
bool __fastcall TForm1::ScaleRead(TObject *Sender)
{
    String buf;
    char *p;
    float ovendryweight, t;
}

```

```

int i, j;
bool c;

c = false;
j = 0;
Delay(10);
ReInputBuf:
if(operationFLG == false) return false;

Comp->InputLen =21;
Comp->InBufferCount =0;

buf.sprintf("5R\r\n");
Comp->RTSEnable = true;
Delay(5);
OutputComp(Comp, buf);
Delay(50);
Comp->RTSEnable = false;

Delay(300);

buf = "";
i = 0;
ReInBuf:
Delay(30);
if (Comp->InBufferCount >= 21)
{
    buf = InputComp(Comp);
    Memo1->Lines->Add(buf);
    Comp->InBufferCount = 0;
    Comp->RTSEnable = true;
}
else
{
    i++;
    if(operationFLG == false) return false;
//    Memo1->Lines->Add(IntToStr(Comp->InBufferCount));

    if (i>20)
    {
        if(Comp->InBufferCount)
        {
            j++;
            if (j>3)
            {
                buf = "통신에러 - 저울";
                Memo1->Lines->Add(buf);
                RunFlagP->Color = clRed;

                ErrorState = true;
                ErLab->Caption = buf;
                return false;
            }else
            {
                i = 0;
                goto ReInBuf;
            }
        }
    }
}

```



```

    }
    else
    {
        if(!c) { c=true; goto ReInputBuf; }
        else return false;
    }
}
goto ReInBuf;
}

if (buf.Length() == 0) return false;
if (buf.Length() >= 17)
{
    p = strtok(buf.c_str(), "- ");
    p = strtok(NULL, "- ");
    if (strlen(p)>=7)
    {
        t = atof(p);
        if(t > 0)
        {
            float a, b;
            a = StrToFloat(RWgtE->Text);
            b = StrToFloat(LWgtE->Text);
//            w : a = t : b
            weight = (a*t) / b;
            Panel10->Caption = buf.sprintf("%5.1f", weight).Trim();
        }
    }else return false;
    buf = OvenWE->Text;
    ovendryweight = StrToFloat(buf);
    MC = (((weight)-ovendryweight)/ovendryweight) *100;
//    MC = (((weight*1000)-ovendryweight)/ovendryweight) *100;
    if (MC > MaxMC)
    {
        MaxMC = MC;
        if (MaxMC < 0) MaxMC = 0;
    }
    Panel19->Caption = buf.sprintf("%5.2f", MC).Trim();
    MCKnob->Pointers->Item(1)->Value = MC;
}else return false;

return true;
}

//-----
float __fastcall TForm1::ReadDBT(TObject *Sender, int nID, int nReg)
{
    String buf;
    String val1, val2;
    int i, j;

    j = 0;

ReInputBuf:
    if (operationFLG == false) return -90;

    Comp->InputLen =16;

```

```

Comp->InBufferCount =0;

buf.sprintf("%c%02dDRS,01,%04d\r\n",2,nID, nReg);
Comp->RTSEnable = true;
OutputComp(Comp, buf);
Delay(50);
Comp->RTSEnable = false;

buf = "";

i = 0;
ReInBuf:
Delay(30);
if (operationFLG == false) return -99;
if (Comp->InBufferCount >= 16)
{
    buf = InputComp(Comp);
    Memo1->Lines->Add(buf.SubString(1, buf.Length()-1));
}
else
{
    i++;
    if(i>5)
    {
        if (j>3)
        {
            buf.sprintf("통신검사요 %dR", nID);
            Memo1->Lines->Add(buf);
            RunFlagP->Color = clRed;

            ErrorState = true;
            ErLab->Caption = buf;
            return -91;
        }
        j++;
        goto ReInputBuf;
    }
}
//    PV1->Color = clBlue;
//    goto ReInBuf;
}
//    Comp->InBufferCount =0;

if (buf.Length() == 0) return -92;

val1 = buf.SubString(4, 6);

if (val1 == "DRS,OK")
    buf.sprintf("%5.1f", HexToDig(buf.SubString(11, 4)));
else
    buf = "-99";

return StrToFloat(buf);
}

//-----
void __fastcall TForm1::NX9Write(int addr, float d)

```

```

{
    String s, buf, buf2;
    int i, j;

    s = IntToHex((int)(d*10), 4);
    // s = IntToHex((int)d, 4);
    j = 0;
ReWrite :
    if (operationFLG == false) return ;
    Comp->InBufferCount =0;
    Comp->InputLen = 10;

    buf.sprintf("%c%02dDWS,02,0300,0001,%s\r\n",2,addr, s);
    Comp->RTSEnable = true;
    OutputComp(Comp, buf);    Delay(50);
    Comp->RTSEnable = false;

    buf2 = "";
    i = 0;
ReInput :
    Delay(30);

    if (operationFLG == false) return ;
    if (Comp->InBufferCount >= 10)
    {
        buf2 = InputComp(Comp);
        Memo1->Lines->Add(buf2.SubString(1, buf2.Length()-1));
    }else
    {
        if(i>5)
        {
            if(j>3)
            {
                ErLab->Caption = buf2.sprintf("통신 검사요 %dW",addr);
                RunFlagP->Color = clRed;

                ErrorState = true;
                Memo1->Lines->Add(buf2);
                return ;
            }
            j++;
            goto ReWrite;
        }
        i++;
        goto ReInput;
    }
}

//-----
void __fastcall TForm1::WindSetValue(TObject *Sender)
{
    int wind_freq;
    String buf;
    int i, zent, j, c;

    if (operationFLG == false) return ;

```

```

Delay(10);

wind = Swind;
wind_freq = (int)((-9.1267*wind*wind*wind)+(27.836*wind*wind)+(6.2033*wind)+9.0552);

if (counterflow == true) buf.sprintf("C%c%c04%02d%04d",3, 1, 3, wind_freq*10); // Addr = 4;
else
    buf.sprintf("C%c%c04%02d%04d",3, 1, 1, wind_freq*10);

j = 0;
c = 0;
ReInputBuf:
Comp->InBufferCount =0;
Comp->InputLen = 11;

zcnt = 0;
Comp->RTSEnable = true;
OutputComp(Comp, buf);
OutputComp(Comp, buf);
Delay(10);
Comp->RTSEnable = false;

i=0;
ReInBuf:
Delay(30);
if (Comp->InBufferCount >= 11)
{
    buf = InputComp(Comp);
    if(buf!="") Memo1->Lines->Add(buf);

    buf.sprintf("  %3.1f", wind);
    Panel13->Caption = buf;
    WindKnob->Pointers->Item(1)->Value = wind;
}
else
{
    i++;
//    Memo1->Lines->Add(IntToStr(Comp->InBufferCount));
    if(Comp->InBufferCount==0) zcnt++;

    if(zcnt > 3)
    {
        c++;
        if(c>3) return ;
        zcnt = 0;
        goto ReInputBuf;
    }
    if (i>5)
        return ;
    goto ReInBuf;
}
}

//-----
int __fastcall LoadSchedule(float *dt, float *wt, float *wd, int opt)
{

```

```

int i, j;
long s;
float mc;

if (opt == 1)    s = floor((double)(Now()-StTime)*24);
else if(opt == 2) s = floor((double)(Now()-StTime)*24*60);

if(WMCCB->Checked) mc = RMC;
else                mc = MC;

*dt = Sche[0].DBT;
*wt = Sche[0].WBT;
*wd = Sche[0].wind;
if(opt == 0)
{
    for (i=0;i<nSchedule;i++)
    {
        if ((Sche[i].st >= mc) &&(Sche[i].et < mc))
        {
            *dt = Sche[i].DBT;
            *wt = Sche[i].WBT;
            *wd = Sche[i].wind;
            break;
        }
    }
    if(i>=nSchedule)
    {
        *dt = Sche[i-1].DBT;
        *wt = Sche[i-1].WBT;
        *wd = Sche[i-1].wind;
    }
}
else
{
    if(opt == 3)
    {
    }
    else
    {
        for (i=0;i<nSchedule;i++)
        {
            if ((Sche[i].st <= s) &&(Sche[i].et > s))
            {
                *dt = Sche[i].DBT;
                *wt = Sche[i].WBT;
                *wd = Sche[i].wind;
                break;
            }
        }
        if(i>=nSchedule)
        {
            *dt = Sche[i-1].DBT;
            *wt = Sche[i-1].WBT;
            *wd = Sche[i-1].wind;
        }
    }
}

```

```

    }
}
return i;
}

//-----
void __fastcall TForm1::StepControl(TObject *Sender)
{
    String buf;
    int schpos;
    int i;

    if (operationFLG == false) return ;
//    if(InLoop) return ;

    InLoop = true;
    ErLab->Caption = "";
    if(!ErrorState) Memo1->Clear();
    RunFlagP->Color = clYellow;

    if(WeightCB->Checked)
    {
        if(!ScaleRead(Sender))
        {
            Delay(100);
            Memo1->Lines->Add(InputComp(Comp)+"----Weight Error");
            RunFlagP->Color = clBlue;
            Comp->InBufferCount = 0;
            Comp->RTSEnable = true;
        }
    }
    if (operationFLG == false) return ;
//    해당 스케줄 읽어오기(건구, 습구, 풍속)
    if (CtlFLG==true)
    {
        schpos = LoadSchedule(&SDBT, &SWBT, &Swind, ScheduleType);
        ScTable->Row = schpos;
    }

    if(WindCB->Checked)
        WindSetValue(Sender);

    if (operationFLG == false) return ;
//    건구 온도, 습구 온도 설정

    NX9Write(1, SDBT);

    if(WBTCB->Checked)
        NX9Write(2, SWBT);
//*****
    float V;

    V = ReadDBT(Sender, 1, 1); if(V>=0) DBT=V;
    if (operationFLG == false) return ;
    Panel3->Caption = buf.sprintf("%5.1f", DBT).Trim();
}

```

```

V = ReadDBT(Sender, 1, 2); if(V>=0) SDBT=V;
if (operationFLG == false) return ;
Panel2->Caption = buf.sprintf("%5.1f", SDBT).Trim();

if(WBTCB->Checked)
{
    V = ReadDBT(Sender, 2, 1); if(V>=0) WBT=V;
    if (operationFLG == false) return ;
    Panel8->Caption = buf.sprintf("%5.1f", WBT).Trim();

    V = ReadDBT(Sender, 2, 2); if(V>=0) SWBT=V;
    if (operationFLG == false) return ;
    Panel9->Caption = buf.sprintf("%5.1f", SWBT).Trim();
}else
{
    WBT = 50;
    SWBT = 50;
}
if(S1C->Checked)
{
    V = ReadDBT(Sender, 3, 1); if(V>=0) IDBT=V;
    if (operationFLG == false) return ;
    S1UP->Caption = buf.sprintf("%5.1f", IDBT).Trim();
}
DBTKnob->Pointers->Item(1)->Value = DBT;
DBTKnob->Pointers->Item(2)->Value = SDBT;

WBTKnob->Pointers->Item(1)->Value = WBT;
WBTKnob->Pointers->Item(2)->Value = SWBT;

// RH, EMC 계산
EMC = DBT-WBT;
if (EMC < 0) EMC = 0;
RH = fRH(DBT, EMC);   buf.sprintf("%4.1f", RH);   Panel15->Caption = buf.Trim();
EMC = fEMC(DBT, RH);   buf.sprintf("%4.1f", EMC); Panel17->Caption = buf.Trim();

//   TextMemo->Lines->Append("-----");
ctrlcount++;

ErrorState = false;
RunFlagP->Color = clBlack;

Delay(100);
Memo1->Lines->Add("Remain Message : " + InputComp(Comp));
Delay(100);

InLoop = false;
}

//-----
float CalcTemp(float S[150])
{
    int i, j;
    float sum;

```

```

float avr;
float S1[150];
float t;

for(i=0;i<150;i++)
    S1[i] = S[i];

for(i=0;i<150;i++)
    for(j=0;j<150;j++)
        if(S1[i] > S1[j])
            {
                t = S1[i];
                S1[i] = S1[j];
                S1[j] = t;
            }

for(i=30, sum=0;i<120;i++)
    sum += S1[i];
avr = sum / 90;

return avr;
}

//-----
void __fastcall TForm1::CRTControl(TObject *Sender)
{
    String buf;
    int schpos, i;
    float V, t;
    float DD, ID;

    static float SDBT1 = 74;
    static float SWBT1 = 74;
    static float SIDBT = 74;

    if (operationFLG == false) return ;
    if(InLoop) return ;

    InLoop = true;
    ErLab->Caption = "";
    if(!ErrorState) Memo1->Clear();
    RunFlagP->Color = clYellow;

    if (operationFLG == false) return ;
    // 해당 스케줄 읽어오기(진구, 습구, 풍속)
    V = ReadDBT(Sender, 1, 1); if(V>=0) DBT=V;
    if (operationFLG == false) return ;
    Panel3->Caption = buf.sprintf("%5.1f", DBT).Trim();

    V = ReadDBT(Sender, 2, 1); if(V>=0) WBT=V;
    if (operationFLG == false) return ;
    Panel8->Caption = buf.sprintf("%5.1f", WBT).Trim();

    V = ReadDBT(Sender, 3, 1); if(V>=0) IDBT=V;
    if (operationFLG == false) return ;
    S2UP->Caption = buf.sprintf("%5.1f", IDBT).Trim();
}

```



```

i = ctrlcount % 150;
DTEMP[i] = DBT;
ITEMP[i] = IDBT;

if (CtlFLG==true)
{
    if(ctrlcount > 150)
    {
        DD = CalcTemp(DTEMP);
        ID = CalcTemp(ITEMP);

        if(MC <= 9) reachlast = true;

        if(reachlast)
        {
            SDBT = 0;
            SWBT = 0;
        }
        else
        {
            if(SDBT1 >= 92)
            {
                SDBT1 = 92;
            }
            else
            {
                if(ID-SIDBT >= 1)
                {
                    StTime25 = Now();
                    limittime = true;

                    SDBT1+= 2;
                    SWBT1--;
                    SIDBT+= 2;
                }
                if(limittime)
                {
                    t = (double)(Now()-StTime25)*24;
                    if(t >=5)
                    {
                        StTime25 = Now();

                        SDBT1+= 2;
                        SWBT1--;
                        SIDBT+= 2;
                    }
                }
            }
        }
    }
}

SDBT = SDBT1;

```

```

SWBT = SWBT1;

if (operationFLG == false) return ;
// 건구 온도, 습구 온도 설정

Panel2->Caption = buf.sprintf("%5.1f", SDBT).Trim();
Panel9->Caption = buf.sprintf("%5.1f", SWBT).Trim();

NX9Write(1, SDBT);
NX9Write(2, SWBT);
//*****

DBTKnob->Pointers->Item(1)->Value = DBT;
DBTKnob->Pointers->Item(2)->Value = SDBT;

WBTKnob->Pointers->Item(1)->Value = WBT;
WBTKnob->Pointers->Item(2)->Value = SWBT;

// RH, EMC 계산
EMC = DBT-WBT;
if (EMC < 0) EMC = 0;
RH = fRH(DBT, EMC);  buf.sprintf("%4.1f", RH);  Panel15->Caption = buf.Trim();
EMC = fEMC(DBT, RH);  buf.sprintf("%4.1f", EMC); Panel17->Caption = buf.Trim();

//  TextMemo->Lines->Append("-----");
ctrlcount++;

ErrorState = false;
RunFlagP->Color = clBlack;

Delay(100);
Memo1->Lines->Add("RM : " + InputComp(Comp));
Delay(100);

InLoop = false;
}

//-----
void __fastcall TForm1::CRTControl2(TObject *Sender)
{
    String buf;
    int schpos;
    int i;
    float EMC;
    long s, s1;
    float swbd, t;

    Comp->InBufferCount =0;

    if (operationFLG == false) return ;
    Panel36->Color = clRed;
    ScaleRead(Sender);

    if (operationFLG == false) return ;
    if (CtlIFLG==false)

```

```

{
    SDBT = StrToFloat(TDBTE->Text);
    SWBT = StrToFloat(TWBTE->Text);
    Swind = StrToFloat(TWNDE->Text);
}

Delay(100);
// 온도, 습도
NX9ReadM(&DBT, &SDBT, Panel3, Panel2, Comp, 1, 280); if (operationFLG == false) return ;
DBTKnob->Pointers->Item(1)->Value = DBT;
DBTKnob->Pointers->Item(2)->Value = SDBT;

// Label79->Caption = IntToStr(Comp->InBufferCount);
Panel36->Color = clGreen;
NX9ReadM(&WBT, &SWBT, Panel8, Panel9, Comp, 2, 280); if (operationFLG == false) return ;

WBTKnob->Pointers->Item(1)->Value = WBT;
WBTKnob->Pointers->Item(2)->Value = SWBT;

// 건구 온도, 습구 온도 설정
Delay(100);
s = floor((double)(Now()-StTime)*24);

if((MC < 25) && (flag25 == false))
{
    if(MC > -30)
    {
        StTime25 = Now();
        flag25 = true;
    }
}

// 건구 설정
if(MC >= 25) SDBT = 55;
else
{
    s1 = floor((double)(Now()-StTime25)*24);
    SDBT = 55 + (0.6*s1);
}

if (SDBT > 90) SDBT = 90;

// 습구 설정
if(MC >= 40)
{
    t = 80.5 - 0.5*s;
    if(t < 55.5) t = 55.5;

    swbd = fWBD(SDBT, t);
    SWBT = SDBT - swbd;
    if(SWBT < 44.8)
        SWBT = 44.8;
}
if((MC < 40) && (MC >=25))
    SWBT = 44.8; // 55oC - 55.5%

if(MC < 25)

```

```

    {
        t = 55.5 - 0.5*s1;
        if (t < 22) t = 22;

        swbd = fWBD(SDBT, t);
        SWBT = SDBT - swbd;
    }

    NX9Write(1, SDBT);
    NX9Write(2, SWBT);

    Panel36->Color = clBlue;

// RH, EMC 계산
    EMC = DBT-WBT;
    if (EMC < 0) EMC = 0;
    RH = fRH(DBT, EMC);   buf.sprintf("%4.1f", RH);   Panel15->Caption = buf.Trim();
    EMC =fEMC(DBT, RH);   buf.sprintf("%4.1f", EMC); Panel17->Caption = buf.Trim();

    Panel36->Color = clBlack;
//   TextMemo->Lines->Append("-----");
    ctrlcount++;

}

//-----

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    switch (ScheduleType)
    {
        case 0 :
        case 1 :
        case 2 : StepControl(Sender);
                break;
        case 3 : CRTControl(Sender);
                break;
        case 4 : CRTControl2(Sender);
                break;
    }
}

// 작동 버튼 -----
void __fastcall TForm1::StartBtnClick(TObject *Sender)
{
    FILE *fp;
    String buf;
    TIniFile *ini;

    DSaveFname = DSaveFnameE->Text;
    fp = fopen(DSaveFname.c_str(), "at");
    fclose(fp);

    if (operationFLG == true)
    {

```

```

// Panel20->Visible = true;
// Timer5->Enabled = true;
return;
}
MC = StrToFloat(InitMCE->Text);

counterflowtime = StrToInt(Edit14->Text);

// Timer4->Interval = counterflowtime * 60* 60* 1000;
// Timer4->Interval = counterflowtime * 60* 1000;

// if (counterflowtime == 0) Timer4->Enabled = false;
// else Timer4->Enabled = true;

if (count != 0)
{
    TextRE->Clear();
    Series1->Clear();
    Series2->Clear();
    Series3->Clear();
    Series4->Clear();
    Series5->Clear();
    MChart->BottomAxis->Maximum = 6;
}
count = 0;
ctrlcount = 0;

operationFLG = true;
CtlFLG = true;
reachlast = false;
limittime = false;

HeadTitle =Edit2->Text + " " + Edit5->Text +"x"+Edit6->Text +"x"+Edit7->Text+"(mm3)";
Caption = "전자동 목재건조시스템 -"+ HeadTitle;

if (!legcstop)
    StTime = Now();
Label5->Caption = DateTimeToStr(StTime);
// Edit3->Text = Label5->Caption;

ini = new TIniFile(ChangeFileExt( Application->ExeName, ".INI" ) );
ini->WriteInteger( "RUNSTOP", "EmcStop", 1 );
ini->WriteDateTime( "RUNSTOP", "StartTime", StTime );

delete ini;

Comp->Interval = 500;
Timer1->Interval = 2000;//StrToInt(DCtrlTE->Text)*1000;
Timer2->Interval = StrToInt(DSaveTE->Text)*1000;

Timer1->Enabled = true;
Timer2->Enabled = true;
Timer3->Enabled = true;

Timer1Timer(Sender);
Timer2Timer(Sender);
}

```

```

// 중지 버튼-----
void __fastcall TForm1::BitBtn25Click(TObject *Sender)
{
    String buf;

    if(operationFLG == true)
    {
        buf.printf("건조중입니다.. 가동을 중지하시겠습니까?");
        if (MessageDlg(buf, mtError, TMsgDlgButtons() << mbYes << mbNo, 0) != mrYes)
            return ;
    }else
        return ;

    operationFLG = false;
    Timer1->Enabled = false;
    Timer2->Enabled = false;
    Timer3->Enabled = false;
    Timer4->Enabled = false;
    RunFlagP->Color = clBlack;

    Delay(500);

    operationFLG = true;
    NX9Write(1, 10.0);
    NX9Write(2, 10.0);

    Delay(80);
    buf.printf("C%c%c0402XXXX",3, 1);

    Comp->RTSEnable = true;
    OutputComp(Comp, buf);
    OutputComp(Comp, buf);
    Delay(10);
    Comp->RTSEnable = false;

    operationFLG = false;

    Panel2->Caption = "10.0";
    Panel9->Caption = "10.0";
    buf.printf("  %3.1f", 0);
    Panel13->Caption = buf;

    DBTKnob->Pointers->Item(2)->Value = 10;
    WBTKnob->Pointers->Item(2)->Value = 10;

    WindKnob->Pointers->Item(1)->Value = 0;

    RestartBtn->Enabled = true;
    StartBtn->Enabled = true;
    InLoop = false;
}

// 스케줄 설정 버튼 -----
void __fastcall TForm1::BitBtn28Click(TObject *Sender)

```

```

{
    int i, j;

    for (i=0;i<ScTable->RowCount;i++)
        for(j=0;j<ScTable->ColCount;j++)
            Form2->ScTable->Cells[j][i] = "";

    for (i=0;i<ScTable->RowCount;i++)
    {
        if (ScTable->Cells[0][i] == "")
        {
            Form2->MainKeyRG->ItemIndex = ScheduleType;
            break;
        }
        for(j=0;j<ScTable->ColCount;j++)
            Form2->ScTable->Cells[j][i] = ScTable->Cells[j][i];
    }

    Form2->Show();
}

//-----
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    String buf;
    TDateTime NowST;
    float fmin;
    long s;
    int m, h, hn, d, dn;

    FILE *f;

    ErLab->Caption = "";

    if ( DSaveFname != "" )
    {
        f = fopen(DSaveFname.c_str(), "at");

        if ( count == 0 )
        {
            NowST = Now();
            fprintf(f, "%s %s \n", HeadTitle, DateTimeToStr(NowST));
            fprintf(f, "      MIN  SDBT  SWBT  DBT  WBT Weight  M C    EMC    RH ITMP
                    MTMP MMC RMC\n");
            StTime = Now();
        }

        //      fmin = floor(fmin);
        fmin = (double)(Now()-StTime)*24*60;
        s = floor((double)(Now()-StTime)*24*60*60);

        d = s / (24*60*60); dn = s % (24*60*60);
        h = dn / (60*60);  hn = dn % (60*60);
        m = hn / 60;      s = hn % 60;
    }
}

```

```

    TimeBuf(sprintf("%d,%02d:%02d",d, h, m);
    GraphOneDraw(Sender);

    buf(sprintf("%8.1f %5.1f %5.1f %5.1f %5.1f %5.1f %5.1f %5.1f %5.1f %5.1f %3.1f %5.1f %5.1f",
                fmin, SDBT, SWBT, DBT, WBT, weight, MC, EMC, RH, IDBT, MTemp, MMC, RMC);
    fputs(buf.c_str(), f);
    fprintf(f, "\n");
    fclose(f);

    TextRE->Lines->Append(buf);
}
count = count + 1;
}

//-----
void __fastcall TForm1::CheckMC(TObject *Sender)
{
    String buf;
    char *p;
    String t11, t12, t21, t22;

    buf = InputComp(MComp);
    Memo1->Lines->Add(buf);

    p = strtok(buf.c_str(), " "); t11 = p;
    p = strtok(NULL, " "); t12 = p;

    p = strtok(NULL, " "); t21 = p;
    p = strtok(NULL, " "); t22 = p;

    if(t11 == "HUM:")
    {
        MMC = StrToFloat(t12)/10;
        MTemp = StrToFloat(t22)/10;
    }

    if(t12 == "HUM:")
    {
        MMC = StrToFloat(t21)/10;
        MTemp = StrToFloat(t11)/10;
    }

    if(t21 == "HUM:")
    {
        MMC = StrToFloat(t22)/10;
        MTemp = StrToFloat(t12)/10;
    }

    if(t22 == "HUM:")
    {
        MMC = StrToFloat(t11)/10;
        MTemp = StrToFloat(t21)/10;
    }

    S5UP->Caption = buf sprintf("%4.1f", MMC).Trim();
    S6UP->Caption = buf sprintf("%4.1f", MTemp).Trim();
}

```



```

if(MMC > 21.1) RMC = 0.5457*MMC+15.603;
else          RMC = 3.3057*MMC-11.866;
S4UP->Caption = buf.sprintf("%.1f", RMC).Trim();
}

//-----
void __fastcall TForm1::Timer3Timer(TObject *Sender)
{
    long s;
    int m, h, hn, d, dn;

    String buf;

    s = floor((double)(Now()-StTime)*24*60*60);

    d = s / (24*60*60); dn = s % (24*60*60);
    h = dn / (60*60);   hn = dn % (60*60);
    m = hn / 60;       s = hn % 60;

    buf.sprintf("%3d일 %02d:%02d:%02d",d, h, m, s);
    PrTimeL->Caption = buf;

    TimeBuf.sprintf("%d,%02d:%02d",d, h, m);

    if(WMCCB->Checked)
        CheckMC(Sender);

// Edit4->Text = buf;
}

// -----
void __fastcall TForm1::Edit9Exit(TObject *Sender)
{
    float i, o, imc;
    String buf;

    i = StrToFloat(InitWE->Text);
    o = StrToFloat(OvenWE->Text);

    imc = ((i-o)/o)*100;

    buf.sprintf("%.2f", imc);

    InitMCE->Text = buf.Trim();
}

//-----
void __fastcall TForm1::BitBtn18Click(TObject *Sender)
{
    PageControl1->ActivePage = TabSheet4;
}

//-----
void __fastcall TForm1::BitBtn20Click(TObject *Sender)
{
    PageControl1->ActivePage = TabSheet2;
}

```

```

    InitMCL->Caption = InitMCE->Text;
    LastMCL->Caption = LastMCE->Text;
}

//-----
void __fastcall TForm1::BitBtn23Click(TObject *Sender)
{
    PageControl1->ActivePage = TabSheet1;
}

//-----
void __fastcall TForm1::SpeedButton1Click(TObject *Sender)
{
    if (SaveDialog1->Execute() == false) return ;
    DSaveFnameE->Text = SaveDialog1->FileName;
}

//-----
void __fastcall TForm1::RestartBtnClick(TObject *Sender)
{
    FILE *fp;
    String buf;
    TIniFile *ini;

    DSaveFname = DSaveFnameE->Text;
    fp = fopen(DSaveFname.c_str(), "wt");
    fclose(fp);
    flag25 = false;

    ini = new TIniFile(ChangeFileExt( Application->ExeName, ".INI" ) );
    ini->WriteInteger( "RUNSTOP", "EmcStop", 1 );
    ini->WriteDateTime( "RUNSTOP", "StartTime", StTime );

    delete ini;

    if (operationFLG == true) return;

    MC = StrToFloat(InitMCE->Text);

    counterflowtime = StrToInt(Edit14->Text);
    counterflow = false;

    Timer4->Interval = counterflowtime * 60* 60* 1000;

    if (counterflowtime == 0) Timer4->Enabled = false;
    else
        Timer4->Enabled = true;

    Caption = "전자동 열기권조시스템 -"+ Edit2->Text + " " + Edit5->Text + "x"
        +Edit6->Text + "x"+Edit7->Text+"(mm3)";

    Label5->Caption = DateTimeToStr(StTime);
    // Edit3->Text = Label5->Caption;

    Timer1->Interval = 1000;
    Timer2->Interval = StrToInt(DSaveTE->Text)*1000*60;

    operationFLG = true;
}

```

```

    CtlFLG = true;

    Timer1->Enabled = true;
    Timer2->Enabled = true;
    Timer3->Enabled = true;
}

//-----
void __fastcall TForm1::MChartClickSeries(TCustomChart *Sender,
    TChartSeries *Series, int ValueIndex, TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    String buf;

    Panel33->Top = Y-40;
    Panel33->Left = X+10;

    SerNameL->Caption = Series->Title;
    XLabL->Caption = Series->XLabel[ValueIndex];
    buf.printf("%.5lf", Series->YValue[ValueIndex]);
    YValL->Caption = buf.Trim();
    Panel33->Font->Color = Series->ValueColor[ValueIndex];
    Panel33->Visible = true;
}

//-----
void __fastcall TForm1::MChartMouseUp(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    Panel33->Visible = false;
}

//-----
void __fastcall TForm1::Timer4Timer(TObject *Sender)
{
    if (counterflow) counterflow = false;
    else counterflow = true;
}

//-----
void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
    PageControll->ActivePage = TabSheet3;
}

//-----
void __fastcall TForm1::BitBtn3Click(TObject *Sender)
{
    Close();
}

//-----
void __fastcall TForm1::Edit14Enter(TObject *Sender)
{
    EditName = dynamic_cast<TEdit *>(Sender)->Name;
}

//-----

```

```

void __fastcall TForm1::BitBtn14Click(TObject *Sender)
{
    FILE *fp;
    char *p;
    char buf[145];
    String TimeBuf;
    float val;
    int i, cnt, lsize;
    long s;
    int m, h, hn, d, dn;

    if (OpenDialog1->Execute() == false) return ;

    TextRE->Lines->LoadFromFile(OpenDialog1->FileName);

    fp = fopen(OpenDialog1->FileName.c_str(), "rt");
    if (fp==NULL)
    {
        // ErMSG1->Caption = "파일을 찾을 수 없습니다.";
        return ;
    }
    fseek(fp, -82, SEEK_END);
    lsize = ftell(fp)/62;

    fseek(fp, 0, SEEK_SET);
    fgets(buf, 140, fp);
    fgets(buf, 140, fp);

    cnt = 0;
    Series1->Clear();
    Series2->Clear();
    Series3->Clear();
    Series4->Clear();
    Series5->Clear();

    MChart->BottomAxis->Maximum = lsize;

    while(!feof(fp))
    {
        fgets(buf, 140, fp);
        if (feof(fp)) break;

        p = strtok(buf, " ");
        TimeBuf = p;

        s = StrToFloat(TimeBuf);
        d = s / (24*60); dn = s % (24*60);
        h = dn / 60;
        m = dn % 60;

        TimeBuf.sprintf("%d,%02d:%02d",d, h, m);

        p = strtok(NULL, " ");
        p = strtok(NULL, " ");
        p = strtok(NULL, " ");

        if (cnt > MChart->BottomAxis->Maximum)
            MChart->BottomAxis->Maximum = MChart->BottomAxis->Maximum+2;

        val = atof(p);
        Series1->Add(val, TimeBuf, clRed);
    }
}

```

```

        p = strtok(NULL, " "); val = atof(p);
        Series2->Add(val, TimeBuf, clBlue);

        p = strtok(NULL, " "); //Weight
        p = strtok(NULL, " "); val = atof(p); //MC
        Series4->Add(val, TimeBuf, clGreen);

        p = strtok(NULL, " "); //EMC

        p = strtok(NULL, " "); val = atof(p); //RH
        Series3->Add(val, TimeBuf, clBlack);
    }

    fclose(fp);
}

//-----
void __fastcall TForm1::BitBtn4Click(TObject *Sender)
{
    if (EditName!="") preComp = FindComponent(EditName);
    else preComp = FindComponent("Edit2");

    if (dynamic_cast<TEdit *>(preComp)->Top < 250) VKeyF->Top = 325;
    else VKeyF->Top = 0;

    VKeyF->Edit1->Text = dynamic_cast<TEdit *>(preComp)->Text;

    VKeyF->Show();
}

//-----
void __fastcall TForm1::BitBtn17Click(TObject *Sender)
{
    if (SaveDialog2->Execute() == false) return ;
    TextRE->Lines->SaveToFile(SaveDialog2->FileName);
}

//-----
void __fastcall TForm1::ScTableDrawCell(TObject *Sender, int ACol,
    int ARow, TRect &Rect, TGridDrawState State)
{
    if (State.Contains(gdSelected))
    {
        ScTable->Canvas->Brush->Style = bsSolid;
        ScTable->Canvas->Brush->Color = clRed;
        ScTable->Canvas->Rectangle( Rect );
        ScTable->Canvas->Font->Color = clWhite;
        ScTable->Canvas->TextOut( Rect.Left+3, Rect.Top+3, ScTable->Cells[ACol][ARow]);
    }
}

//-----
void __fastcall TForm1::Image1DbClick(TObject *Sender)
{
    Memo1->Visible = !Memo1->Visible;
}

//-----
void __fastcall TForm1::BitBtn6Click(TObject *Sender)
{
    String buf;

```

```

int val;

buf = TGrpRangE2->Text.Trim();
if(buf!="") val = 150;
else      val = StrToInt(buf);

MChart->LeftAxis->Maximum = val;
DBTKnob->Axis->Maximum = val;
WBTKnob->Axis->Maximum = val;

buf = TGrpRangE1->Text.Trim();
if(buf!="") val = 0;
else      val = StrToInt(buf);

MChart->LeftAxis->Minimum = val;
//  DBTKnob->Axis->Minimum = val;
//  WBTKnob->Axis->Minimum = val;
}

//-----
void __fastcall TForm1::BitBtn7Click(TObject *Sender)
{
    String buf;
    int val;

    buf = MGrpRangE2->Text.Trim();
    if(buf!="") val = 100;
    else      val = StrToInt(buf);

    MChart->RightAxis->Maximum = val;
    MCKnob->Axis->Maximum = val;

    buf = MGrpRangE1->Text.Trim();
    if(buf!="") val = 0;
    else      val = StrToInt(buf);

    MChart->RightAxis->Minimum = val;
//  MCKnob->Axis->Minimum = val;
}

//-----
void __fastcall TForm1::WMCCBClick(TObject *Sender)
{
    CheckBox5->Checked = WMCCB->Checked;
    CheckBox6->Checked = WMCCB->Checked;
}

//-----
void __fastcall TForm1::FormPaint(TObject *Sender)
{
    if (egcstop)
        StartBtnClick(Sender);
}
//-----

```

```

//*****
//SchSet.CPP
//*****
#include <vcl.h>
#pragma hdrstop

#include "SchSet.h"
#include "Math.h"
#include "Main.h"
#include "Keyboard.h"
#include "stdio.h"

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"

void __fastcall TypeSelClick(TObject *Sender);
void __fastcall FormCreate(TObject *Sender);
void __fastcall DBGrid1CellClick(TColumn *Column);
void __fastcall EstimatBClick(TObject *Sender);
void __fastcall BSGClick(TObject *Sender);
void __fastcall BitBtn3Click(TObject *Sender);
void __fastcall BitBtn1Click(TObject *Sender);
void __fastcall EndButClick(TObject *Sender);
void __fastcall BitBtn7Click(TObject *Sender);
void __fastcall BitBtn8Click(TObject *Sender);
void __fastcall NewSchBClick(TObject *Sender);
void __fastcall BitBtn5Click(TObject *Sender);
void __fastcall BitBtn4Click(TObject *Sender);
void __fastcall BitBtn9Click(TObject *Sender);
void __fastcall BitBtn2Click(TObject *Sender);
void __fastcall MainKeyRGClick(TObject *Sender);
void __fastcall ScTableKeyPress(TObject *Sender, char &Key);
void __fastcall M8tEnter(TObject *Sender);
void __fastcall DBNavigator1Click(TObject *Sender,TNavigateBtn Button);
void __fastcall FormShow(TObject *Sender);
void __fastcall DBGrid2CellClick(TColumn *Column);
void __fastcall HSchCodeCBChange(TObject *Sender);
void __fastcall BitBtn10Click(TObject *Sender);
void __fastcall HSchCodeOut(int row, String S1, String S2, String S3);
void __fastcall ScTableSelectCell(TObject *Sender, int ACol, int ARow, bool &CanSelect);
void __fastcall DBEdit9Enter(TObject *Sender);
void __fastcall BitBtn6Click(TObject *Sender);

struct StartMC
{
    int stmc, edmc;
};

struct Schedul
{
    float MC1, MC2 , SDBT ,WBD, SRH;
};

StartMC St_MC[7] =
{
    {40, 30},
    {60, 40},
    {80, 60},
    {100, 80},
    {120, 100},

```

```

    {140, 120},
    {260, 140}
};

TForm2 *Form2;
Schedul SSche[11];
extern Schedule Sche[40];
extern int nSchedule;
extern int ScheduleType;
extern CRTS crtsch;

int saveflag;

extern TComponent *preComp;
String EditName1;
int scol, srow;

extern float __fastcall fRH(float DBT, float WBD);
extern float __fastcall fEMC(float T, float RH);
extern float fWBD(float DBT, float RH);

//-----
__fastcall TForm2::TForm2(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm2::TypeSelClick(TObject *Sender)
{
    String nodeStr, S1;
    char *p;

    nodeStr = TypeSel->Items[TypeSel->SelectedItem]->FullPath;

    p = strtok(nodeStr.c_str(), "/");
    p = strtok(NULL, "/");

    if (!p)
    {
        DB_QT->Filtered = False;
        return ;
    }else
    {
        DB_QT->Filtered = False;

        S1.printf("(BRA1 = '%s')", p);
        DB_QT->Filter = S1;
        p = strtok(NULL, "/");
        if(p)
        {
            S1 = DB_QT->Filter + S1.printf(" and (BRA2 = '%s')", p);
            DB_QT->Filter = S1;
        }
    }
    DB_QT->Filtered = True;
}
//-----
void __fastcall TForm2::FormCreate(TObject *Sender)
{
    TypeSel->FullExpand();
}

```



```

ScTab->Cells[0][0] = "IMC";
ScTab->Cells[1][0] = "FMC";
ScTab->Cells[2][0] = "DBT";
ScTab->Cells[3][0] = "WBD";
ScTab->Cells[4][0] = "RH";
ScTab->Cells[5][0] = "EMC";
ScTab->Cells[6][0] = "DAYS";
ScTab->Cells[7][0] = "%/DAY";

saveflag = 0;
}
//-----

void __fastcall TForm2::DBGrid1CellClick(TColumn *Column)
{
    String s;

    s = DBGrid1->SelectedField->AsString;
    if (s=="")
    {
        SchCodeE->Text= "";
    }
    else
    {
        if (s[1] == 'T')    SchCodeE->Text= s;
        else                SchCodeE->Text= "";
    }
    EditName1 = "DBGrid1";
}
//-----

void __fastcall TForm2::EstimatBClick(TObject *Sender)
{
    float M[8], D[8], T[8], EMC, DAY, RH, RATE;
    float gb, gm, L, MC, tmpv;
    float mm, mg, mtp;
    float A, B, BS, BT1, BT2, TDay;
    String stmp;
    int i, j;

    for (i=1;i<ScTab->RowCount;i++)
        for(j=0;j<ScTab->ColCount;j++)
            ScTab->Cells[j][i] = "";

// Basic Specific Gravity Estimation
if (BSG->ItemIndex == 0 )
    gb = StrToFloat(GBt->Text);
else
{
    gm = StrToFloat(GMt->Text);
    MC = StrToFloat(Mt->Text);
    gb = gm / (1+ 0.265*gm*((30-MC)/30));
}

M[8] = StrToFloat(M8t->Text);
L = StrToFloat(Lt->Text);
M[0] = StrToFloat(M0t->Text);

// Green Moisture content Estimation

```

```

mm = 100*(1.48-gb)/ (1.48*gb);
tmpv = gb*gb;
mg = mm*(1-exp(-1+(1.54*gb)-(3.23*tmpv)));
if (M[0] < mg ) M[0] = mg;

// Initial DBT And WBD Calculation
T[0] = 1/(0.0115 + 0.0167 * gb);
D[0] = 1/(0.0832 + 0.509 * gb);

// DBT And WBD increment Calculation
A = 1/(0.0011+0.00124*D[0]);
B = 0.14 - 0.0458* log(D[0]);

for (i=1;i<8;i++)
{
M[i] = 45-5*(i-1);
if (M[i] <= M[8])
M[i] = M[8];
}

for (i=1;i<8;i++)
{
if (M[i]>30 ) T[i] = T[0];
else T[i] = T[0]+5.6*((30-M[i])/5+1);

D[i] = A*exp(-B*M[i]);
if((T[i]-D[i])<=32 )
D[i] = T[i]-32;

if (D[i] >= 27.8 )
D[i] = 27.8;
}

if (gb<0.75 ) T[7] = 82;
else T[7] = 71;

TDay = 0;
// Drying Time and Rate Calculation
for (i=0; i<8;i++)
{
if (M[8] >= M[i]) break;

RH = fRH(T[i], D[i]);
EMC = fEMC(T[i], RH);

if (M[i+1]<EMC) M[i+1] = ceil(EMC);

BS = 1.42 + (18.2/gb);
BT1 = 0.0575+0.00142*exp(20.4-5132/(49+273));
BT2 = 0.0575+0.00142*exp(20.4-5132/(T[i]+273));

DAY = (pow(L, 1.52)*-1)* log((M[i+1]-EMC) / (M[i]-EMC)) *(BT1/BT2)/BS;
RATE = (M[i]-M[i+1])/DAY;
TDay = TDay +DAY;

stmp(sprintf("%3.0f",M[i]); ScTab->Cells[0][i+1] = stmp.Trim();
stmp(sprintf("%3.0f",M[i+1]); ScTab->Cells[1][i+1] = stmp.Trim();
stmp(sprintf("%3.1f",T[i]); ScTab->Cells[2][i+1] = stmp.Trim();

```

```

        stmp.sprintf("%3.1f",D[i]);    ScTab->Cells[3][i+1] = stmp.Trim();
        stmp.sprintf("%3.1f",RH);    ScTab->Cells[4][i+1] = stmp.Trim();
        stmp.sprintf("%4.1f",EMC);   ScTab->Cells[5][i+1] = stmp.Trim();
        stmp.sprintf("%5.1f",DAY);   ScTab->Cells[6][i+1] = stmp.Trim();
        stmp.sprintf("%5.1f",RATE);  ScTab->Cells[7][i+1] = stmp.Trim();
    }
    stmp.sprintf("%5.2f",TDay);
    ScTab->Cells[6][i+1] = stmp;
    MainKeyRG->ItemIndex = 0;
}
//-----

void __fastcall TForm2::BSGClick(TObject *Sender)
{
    if (BSG->ItemIndex == 0)
    {
        Panel8->Visible = True;
        GBt->Visible = True;
        Panel9->Visible = False;
        Panel10->Visible = False;
        GMt->Visible = False;
        Mt->Visible = False;
    }
    else
    {
        Panel9->Visible = True;
        Panel10->Visible = True;
        GMt->Visible = True;
        Mt->Visible = True;
        Panel8->Visible = False;
        GBt->Visible = False;
    }
}
//-----

void __fastcall TForm2::BitBtn3Click(TObject *Sender)
{
    String ScSym, Tmsc, Emsc, EmscA, EmscN;
    int ls, i, j, k, emType, WoodType;
    int  sst, sed,          // Start MC
        tst, ted,          // DBT  }
        wst, wed ;        // wood }
    int tdo_c, r;
    float wdo_c;

    ScSym = SchCodeE->Text;
    if (SchCodeE->Text=="") return ;

    WoodType = 1;
    if (DB_QT->FieldByName("BRA2")->AsString == "침엽수") WoodType = 0;
    else
        WoodType = 1;

    RHT->Active = False;
    if (WoodType == 0 ) RHT->TableName = "SoftS.DBF";
    else
        RHT->TableName = "HardS.DBF";
}

```

```

RHT->Active = True;

ls = ScSym.Length();

if (ls == 4 )
{
    Tmsc = ScSym.SubString(1, 2);
    Emsc = ScSym.SubString(3, 2);
}
else if ( ls == 5 )
{
    Tmsc = ScSym.SubString(1, 3);
    Emsc = ScSym.SubString(4, 2);
}

EmscA = Emsc.SubString(1, 1);
EmscN = "Z"+Emsc.SubString(2, 1);

emType = Emsc[1]-65;
sst = St_MC[emType].stmc;
sed = St_MC[emType].edmc;

TempT->First();
RHT->First();

tst = TempT->FieldByName("S_MC")->AsInteger;
ted = TempT->FieldByName("E_MC")->AsInteger;
tdo_c = TempT->FieldByName(Tmsc)->AsInteger;

wst = RHT->FieldByName(EmscA)->AsInteger;
wed = RHT->FieldByName(EmscA+"_")->AsInteger;
wdo_c = RHT->FieldByName(EmscN)->AsFloat;

SSche[0].MC1 = sst;
SSche[0].MC2 = wed;
SSche[0].SDBT = tdo_c;
SSche[0].WBD = wdo_c;

if ( emType == 0 )
{
    RHT->Next();
    TempT->Next();

    tst = TempT->FieldByName("S_MC")->AsInteger;
    ted = TempT->FieldByName("E_MC")->AsInteger;
    tdo_c = TempT->FieldByName(Tmsc)->AsInteger;

    wst = RHT->FieldByName(EmscA)->AsInteger;
    wed = RHT->FieldByName(EmscA+"_")->AsInteger;
    wdo_c = RHT->FieldByName(EmscN)->AsFloat;
}

i = 1;
while(1)

```

```

{
    if ( ted < wed )
        RHT->Next();
    else
    {
        if ( ted == wed )
        {
            RHT->Next();
            TempT->Next();
        }
        else
            TempT->Next();
    }
    if ((TempT->Eof) && (RHT->Eof )) break;
    if ( RHT->FieldByName(EmscA)->AsInteger == 0 )
    {
        RHT->Last();
        TempT->Last();
    }

    ted = TempT->FieldByName("E_MC")->AsInteger;
    tdo_c = TempT->FieldByName(Tmsc)->AsInteger;

    wst = RHT->FieldByName(EmscA)->AsInteger;
    wed = RHT->FieldByName(EmscA+"_-")->AsInteger;
    wdo_c = RHT->FieldByName(EmscN)->AsFloat;

    if ( ted < wed ) SSche[i].MC2 = wed;
    else          SSche[i].MC2 = ted;

    SSche[i].MC1 = SSche[i-1].MC2;

    SSche[i].SDBT = tdo_c;
    SSche[i].WBD  = wdo_c;

    if (SSche[i].MC2 == 0) break;

    i = i+1;
}
for (k=0;k<ScTable->RowCount;k++)
    for(j=0;j<ScTable->ColCount;j++)
        ScTable->Cells[j][k] = "";

float RH, EMC;
for(j=1; j<=i+1; j++)
{
    RH = fRH(SSche[j-1].SDBT, SSche[j-1].WBD);
    EMC = fEMC(SSche[j-1].SDBT, RH);

    Tmsc.printf("%3.0f",SSche[j-1].MC1); ScTable->Cells[0][j-1] = Tmsc.Trim();
    if (j==1) ScTable->Cells[0][j-1] = "生";
    Tmsc.printf("%3.0f",SSche[j-1].MC2); ScTable->Cells[1][j-1] = Tmsc.Trim();
    Tmsc.printf("%5.1f",SSche[j-1].SDBT); ScTable->Cells[2][j-1] = Tmsc.Trim();
    Tmsc.printf("%5.1f",SSche[j-1].SDBT - SSche[j-1].WBD);
        ScTable->Cells[3][j-1] = Tmsc.Trim();
}

```

```

        Tmsc.printf("%5.1f",RH);           ScTable->Cells[5][j-1] = Tmsc.Trim();
        Tmsc.printf("%5.1f",EMC);        ScTable->Cells[6][j-1] = Tmsc.Trim();
    }
    MainKeyRG->ItemIndex = 0;
}
//-----

void __fastcall TForm2::BitBtn1Click(TObject *Sender)
{
    int i, j;
    String buf;
    float a, b;

    for (i=0;i<ScTable->RowCount;i++)
        for(j=0;j<ScTable->ColCount;j++)
            ScTable->Cells[j][i] = "";

    for (i=0;i<=7;i++)
    {
        if(ScTab->Cells[0][i+1]== "") break;

        ScTable->Cells[0][i] = ScTab->Cells[0][i+1];
        ScTable->Cells[1][i] = ScTab->Cells[1][i+1];
        ScTable->Cells[2][i] = ScTab->Cells[2][i+1];

        a = StrToFloat(ScTab->Cells[2][i+1]);
        b = StrToFloat(ScTab->Cells[3][i+1]);
        buf.printf("%5.1f", a-b);
        ScTable->Cells[3][i] = buf.Trim();

        ScTable->Cells[5][i] = ScTab->Cells[4][i+1];
        ScTable->Cells[6][i] = ScTab->Cells[5][i+1];
    }
}
//-----

void __fastcall TForm2::EndButClick(TObject *Sender)
{
    int i, j;
    String buf;

    ScheduleType = MainKeyRG->ItemIndex;

    if(ScheduleType == 3)
    {
        crtsch.dDBT0 = StrToFloat(dDBT0E->Text);
        crtsch.dDBT1 = StrToFloat(dDBT1E->Text);
        crtsch.mxDBT = StrToFloat(mxDBTE->Text);

        crtsch.dWBT0 = StrToFloat(dWBT0E->Text);
        crtsch.dWBT1 = StrToFloat(dWBT1E->Text);
        crtsch.mxWBT = StrToFloat(mxWBTE->Text);

        crtsch.stDBT = StrToFloat(Edit12->Text);
    }
}

```

```

Form1->dDBTSL->Caption = dDBT0E->Text;
Form1->dDBTEL->Caption = dDBT1E->Text;
Form1->dDBT0L->Caption = mxDBTE->Text;

Form1->dWBTS�->Caption = dWBT0E->Text;
Form1->dWBTEL->Caption = dWBT1E->Text;
Form1->dWBT0L->Caption = mxWBTE->Text;

Form1->WBTimeL->Caption = Edit12->Text;
}else
{

saveflag = 1;
BitBtn4Click(Sender);
saveflag = 0;
switch (MainKeyRG->ItemIndex)
{
case 0 :
case 3 :
case 4 :
Form1->Label74->Caption = "합수율";
Form1->Label75->Caption = "( % )";
break;
case 1 :
Form1->Label74->Caption = "시 간";
Form1->Label75->Caption = "(hour)";
break;
case 2 :
Form1->Label74->Caption = "시 간";
Form1->Label75->Caption = "(min)";
break;
}
for (i=0;i<ScTable->RowCount;i++)
for(j=0;j<ScTable->ColCount;j++)
Form1->ScTable->Cells[j][i] = "";

for (i=0;i<ScTable->RowCount;i++)
{
if (ScTable->Cells[0][i] == "")
{
nSchedule = i;
ScheduleType = MainKeyRG->ItemIndex;
break;
}
if ((ScTable->Cells[4][i] == "") || (ScTable->Cells[4][i] == " "))
ScTable->Cells[4][i] = "1.0";

for(j=0;j<ScTable->ColCount;j++)
Form1->ScTable->Cells[j][i] = ScTable->Cells[j][i];

if((i==0) && (MainKeyRG->ItemIndex==0)) Sche[i].st = 400;
else Sche[i].st = ScTable->Cells[0][i].ToInt();

Sche[i].et = ScTable->Cells[1][i].ToInt();

Sche[i].DBT = ScTable->Cells[2][i].ToDouble();
Sche[i].WBT = ScTable->Cells[3][i].ToDouble();
Sche[i].wind = ScTable->Cells[4][i].ToDouble();
}
}

```

```

        Sche[i].RH = ScTable->Cells[5][i].ToDouble();
        Sche[i].EMC = ScTable->Cells[6][i].ToDouble();
    }
}

Close();
switch(ScheduleType)
{
    case 0 :
    case 1 :
    case 2 : Form1->SCTNB1->PageIndex = 0; break;
    case 3 : Form1->SCTNB1->PageIndex = 1; break;
    case 4 : Form1->SCTNB1->PageIndex = 2; break;
}
}
//-----

void __fastcall TForm2::BitBtn7Click(TObject *Sender)
{
    int i, j, r, e;

    r = ScTable->Row;
    for(i=ScTable->RowCount-1; i>=r;i--)
        if (ScTable->Cells[1][i] != "")
            break;
    e = i;

    for(i=e; i>=r;i--)
    {
        for(j=0;j<ScTable->ColCount;j++)
            ScTable->Cells[j][i+1] = ScTable->Cells[j][i];
    }
    for(j=0;j<ScTable->ColCount;j++)
        ScTable->Cells[j][r] = "";
}
//-----

void __fastcall TForm2::BitBtn8Click(TObject *Sender)
{
    int i, j, r, e;

    r = ScTable->Row;
    for (i=r;i<ScTable->RowCount;i++)
    {
        for(j=0;j<ScTable->ColCount;j++)
            ScTable->Cells[j][i] = ScTable->Cells[j][i+1];
    }
    for(j=0;j<ScTable->ColCount;j++)
        ScTable->Cells[j][ScTable->RowCount-1] = "";
}
//-----

void __fastcall TForm2::NewSchBClick(TObject *Sender)
{
    int i, j;

    for (i=0;i<ScTable->RowCount;i++)
    {
        for(j=0;j<ScTable->ColCount;j++)

```



```

        ScTable->Cells[j][i] = "";
    }
}
//-----

void __fastcall TForm2::BitBtn5Click(TObject *Sender)
{
    int i, j, r;
    char buf[100];
    char *p;
    String buf2;
    FILE *fp;

    if ( OpenFileDialog1->Execute() == True )
    {
        NewSchBClick(Sender);

        fp = fopen(OpenDialog1->FileName.c_str(), "rt");
        fseek(fp, 0, SEEK_SET);
        fgets(buf, 95, fp);
        MainKeyRG->ItemIndex = atoi(buf);
        i = 0;
        while (!feof(fp))
        {
            fgets(buf, 95, fp);
            if(feof(fp)) break;

            p = strtok(buf, ",");
            if (p == NULL) break;

            for(j=0;j<ScTable->ColCount;j++)
            {
                ScTable->Cells[j][i] = p;
                p = strtok(NULL, ",");
            }
            i++;
        }
        fclose(fp);
    }
}
//-----

void __fastcall TForm2::BitBtn4Click(TObject *Sender)
{
    int i, j;
    String buf, buf2;
    FILE *fp;

    if (saveflag == 0)
    {
        if (SaveDialog1->Execute() == False) return ;
        buf = SaveDialog1->FileName;
    }else
        buf = "init.stb";

    fp = fopen(buf.c_str(), "wt");
    fprintf(fp, "%d\n", MainKeyRG->ItemIndex);
    for (i=0;i<ScTable->RowCount;i++)
    {
        if (ScTable->Cells[0][i] == "") break;

```

```

        buf = "";
        for(j=0;j<ScTable->ColCount;j++)
        {
            buf2 = ScTable->Cells[j][i];
            if (buf2 == "") buf2 = " ";
            buf = buf + buf2 + ",";
        }
        fprintf(fp, "%s\n", buf.c_str());
    }
    fclose(fp);
}

//-----
void __fastcall TForm2::BitBtn9Click(TObject *Sender)
{
    int i;

    for (i=0;i<ScTable->RowCount;i++)
    {
        if (ScTable->Cells[0][i] == "") break;

        ScTable->Cells[4][i] = WindDupE->Text;
    }
}

//-----
void __fastcall TForm2::BitBtn2Click(TObject *Sender)
{
    Close();
}

//-----
void __fastcall TForm2::MainKeyRGClick(TObject *Sender)
{
    switch (MainKeyRG->ItemIndex)
    {
        case 0 :
            Label74->Caption = "합수율";
            Label75->Caption = "( % )";
            break;
        case 1 :
            Label74->Caption = "시 간";
            Label75->Caption = "(hour)";
            break;
        case 2 :
            Label74->Caption = "시 간";
            Label75->Caption = "(min)";
            break;
    }
}

//-----
void __fastcall TForm2::ScTableKeyPress(TObject *Sender, char &Key)
{
    int r, c, flg;
    float t, w, RH, wbd, EMC;
    String buf;

    if (Key == VK_RETURN)
    {
        r = ScTable->Row;
        c = ScTable->Col;
    }
}

```

```

    flg = 0;
    if ((c == 2) || (c == 3))
    {
        if ((ScTable->Cells[2][r] != "") && (ScTable->Cells[3][r] != ""))
        {
            t = StrToFloat(ScTable->Cells[2][r]);
            w = StrToFloat(ScTable->Cells[3][r]);

            if (w > t)
            {
                ScTable->Cells[3][r] = ScTable->Cells[2][r];
                w = t;
            }

            RH = fRH(t, t-w);
            EMC = fEMC(t, RH);

            buf.sprintf("%3.1f",RH);          ScTable->Cells[5][r] = buf.Trim();
            buf.sprintf("%3.1f",EMC);        ScTable->Cells[6][r] = buf.Trim();
            flg = 1;
        }
    }

    if ((c == 2) || (c == 5))
    {
        if (flg == 0)
        {
            if ((ScTable->Cells[2][r] != "") && (ScTable->Cells[5][r] != ""))
            {
                t = StrToFloat(ScTable->Cells[2][r]);
                RH = StrToFloat(ScTable->Cells[5][r]);
                if (RH > 100)
                {
                    ScTable->Cells[5][r] = "100.0";
                    RH = 100.0;
                }

                wbd = fWBD(t, RH);
                EMC = fEMC(t, RH);

                buf.sprintf("%3.1f",t-wbd);    ScTable->Cells[3][r] = buf.Trim();
                buf.sprintf("%3.1f",EMC);      ScTable->Cells[6][r] = buf.Trim();
            }
        }
    }

    if(ScTable->Col == ScTable->ColCount-1)
    {
        ScTable->Col = 0;
        if(ScTable->Row != ScTable->RowCount-1)
            ScTable->Row++;
    }
    else
        ScTable->Col++;
}

}

//-----
void __fastcall TForm2::M8tEnter(TObject *Sender)
{
    EditName1 = dynamic_cast<TEdit *>(Sender)->Name;
}

```

```

//-----
void __fastcall TForm2::DBNavigator1Click(TObject *Sender,
    TNavigateBtn Button)
{
    String nodeStr;
    char *p;

    if (Button == nbInsert)
    {
        nodeStr = TypeSel->Items[TypeSel->SelectedItem]->FullPath;

        p = strtok(nodeStr.c_str(), "/");
        p = strtok(NULL, "/");

        if (!p)
        {
            return ;
        }
        else
        {
            {
                DB_QT->FieldByName("BRA1")->AsString = p;
                p = strtok(NULL, "/");
                if(p) DB_QT->FieldByName("BRA2")->AsString = p;
            }
        }
    }
}

//-----
void __fastcall TForm2::FormShow(TObject *Sender)
{
    TempT->Active = True;
    HiTempT->Active = True;
    RHT->Active = True;
    DB_QT->Active = True;

    HSchCodeCBChange(Sender);
    EditName1 = "WindDupE";
}

//-----
void __fastcall TForm2::DBGrid2CellClick(TColumn *Column)
{
    String s;

    s = DBGrid2->SelectedField->AsString;
    if (s=="")
    {
        HSchCodeCB->Text= "";
    }
    else
    {
        if (s[2] == 'H')    HSchCodeCB->Text= s;
        else              HSchCodeCB->Text= "";
    }
    HSchCodeCBChange(HiSchSG);
    EditName1 = "DBGrid2";
}

//-----
void __fastcall TForm2::HSchCodeOut(int row, String S1, String S2, String S3)
{
    HiSchSG->Cells[0][row] = S1;
    HiSchSG->Cells[1][row] = S2;
    HiSchSG->Cells[2][row] = S3;
}

```

```

}

void __fastcall TForm2::HSchCodeCBChange(TObject *Sender)
{
    String s;
    int i, j;

    s = HSchCodeCB->Text;
    if(s!="")
    {
        if(s[2]=='H')
        {
            for(i=0;i<7;i++)
                for(j=0;j<3;j++)
                    HiSchSG->Cells[j][i] = "";

            switch(s[1])
            {
                case 'A' : HSchCodeOut(0, "17", "104", "82.2");
                           HSchCodeOut(1, "equal.& condit.", "", "");
                           CondL->Caption = "(hour)";
                           break;
                case 'B' : HSchCodeOut(0, "16", "104", "82.2");
                           HSchCodeOut(1, "equal.& condit.", "", "");
                           CondL->Caption = "(hour)";
                           break;
                case 'C' : HSchCodeOut(0, "15", "104", "82.2");
                           HSchCodeOut(1, "equal.& condit.", "", "");
                           CondL->Caption = "(hour)";
                           break;
                case 'D' : HSchCodeOut(0, "24", "116", "82.2");
                           HSchCodeOut(1, "equal.& condit.", "", "");
                           CondL->Caption = "(hour)";
                           break;
                case 'E' : HSchCodeOut(0, "41", "104", "82.2");
                           HSchCodeOut(1, "equal.& condit.", "", "");
                           CondL->Caption = "(hour)";
                           break;
                case 'F' : HSchCodeOut(0, "8", "104", "98.9");
                           HSchCodeOut(1, "24", "104", "96.1");
                           HSchCodeOut(2, "60", "104", "93.3");
                           HSchCodeOut(3, "96", "107", "93.3");
                           HSchCodeOut(4, "Final", "113", "93.3");
                           HSchCodeOut(5, "equal.& condit.", "", "");
                           CondL->Caption = "(hour)";
                           break;
                case 'G' : HSchCodeOut(0, "6", "100", "100");
                           HSchCodeOut(1, "16", "116", "87.8");
                           HSchCodeOut(2, "Final", "116", "76.7");
                           HSchCodeOut(3, "equal.& condit.", "", "");
                           CondL->Caption = "(hour)";
                           break;
                case 'H' : HSchCodeOut(0, "4", "89.9", "98.9");
                           HSchCodeOut(1, "8", "104", "98.9");
                           HSchCodeOut(2, "12", "110", "96.0");
                           HSchCodeOut(3, "18", "110", "93.3");
            }
        }
    }
}

```

```

        HSchCodeOut(4, "24", "110", "87.8");
        HSchCodeOut(5, "Final", "110", "82.2");
        HSchCodeOut(6, "equal.& condit.", "", "");
        CondL->Caption = "(hour)";
        break;
    case 'I' : HSchCodeOut(0, "warm-2", "-", "98.9");
        HSchCodeOut(1, ">30%", "110", "97.8");
        HSchCodeOut(2, "<30%", "110", "88.9");
        HSchCodeOut(3, "condit.", "87.8", "82.2");
        CondL->Caption = "(%)";
        break;
    case 'J' : HSchCodeOut(0, "warm-2", "-", "98.9");
        HSchCodeOut(1, ">35%", "113", "93.3");
        HSchCodeOut(2, "35%-20%", "116", "87.8");
        HSchCodeOut(3, "<20%", "118", "82.2");
        HSchCodeOut(4, "condit.", "87.8", "82.2");
        CondL->Caption = "(%)";
        break;
    case 'K' : HSchCodeOut(0, "warm-3", "-", "98.9");
        HSchCodeOut(1, "hold-1/2", "116", "98.9");
        HSchCodeOut(2, "Final", "116", "93.3");
        HSchCodeOut(3, "condit.", "104", "100");
        CondL->Caption = "(%)";
        break;
    case 'N' : HSchCodeOut(0, "28", "110", "85.0");
        HSchCodeOut(1, "equal.& condit.", "", "");
        CondL->Caption = "(hour)";
        break;
    case 'O' : HSchCodeOut(0, "17", "107", "87.8");
        HSchCodeOut(1, "21", "118", "87.8");
        HSchCodeOut(2, "24", "96.1", "82.2");
        HSchCodeOut(3, "equal.& condit.", "", "");
        CondL->Caption = "(hour)";
        break;
    case 'P' : HSchCodeOut(0, "42", "116", "96.1");
        HSchCodeOut(1, "equal.& condit.", "", "");
        CondL->Caption = "(hour)";
        break;
    case 'Q' : HSchCodeOut(0, "54", "113", "82.2");
        HSchCodeOut(1, "58", "steam", "-");
        HSchCodeOut(2, "62", "113", "82.2");
        HSchCodeOut(3, "66", "steam", "-");
        HSchCodeOut(4, "90", "113", "82.2");
        HSchCodeOut(5, "equal.& condit.", "", "");
        CondL->Caption = "(hour)";
        break;
    }
}
}
}

//-----
void __fastcall TForm2::BitBtn10Click(TObject *Sender)
{
    if (EditName1!="") preComp = FindComponent(EditName1);
    else                preComp = FindComponent("WindDupE");

    if(preComp->Tag == 0)
    {

```

```

        if (dynamic_cast<TEdit *>(preComp)->Top < 250) VKeyF->Top = 325;
        else
            VKeyF->Top = 0;

        VKeyF->Edit1->Text = dynamic_cast<TEdit *>(preComp)->Text;
    }
    if(preComp->Tag == 11)
    {
        if (dynamic_cast<TDBEdit *>(preComp)->Top < 250) VKeyF->Top = 325;
        else
            VKeyF->Top = 0;

        VKeyF->Edit1->Text = DB_QT->FieldByName(dynamic_cast<TDBEdit
*>(preComp)->DataField)->AsString;
    }

    if(preComp->Tag == 21) VKeyF->Edit1->Text = DBGrid2->SelectedField->AsString;
    if(preComp->Tag == 22) VKeyF->Edit1->Text = DBGrid1->SelectedField->AsString;

    if(preComp->Tag == 13)
    {
        VKeyF->Top = 325;
        VKeyF->Edit1->Text = dynamic_cast<TStringGrid *>(preComp)->Cells[scol][srow];
    }
    VKeyF->Show();
}
//-----
void __fastcall TForm2::ScTableSelectCell(TObject *Sender, int ACol,
    int ARow, bool &CanSelect)
{
    EditName1 = "ScTable";
    scol = ACol;
    srow = ARow;
}
//-----
void __fastcall TForm2::DBEdit9Enter(TObject *Sender)
{
    EditName1 = dynamic_cast<TDBEdit *>(Sender)->Name;
}
//-----
void __fastcall TForm2::BitBtn6Click(TObject *Sender)
{
    String s;

    s = HSchCodeCB->Text;

    if(s[1] >='I' && s[1] <= 'K') MainKeyRG->ItemIndex = 0;
    else
        MainKeyRG->ItemIndex = 1;

    MainKeyRGClick(Sender);
    MainKeyRG->ItemIndex = 0;
}
//-----
void __fastcall TForm2::CreateParams( TCreateParams &Params )
{
    TForm::CreateParams( Params ); // to call basic class
    Params.Style &= ~WS_CAPTION; // remove Caption
}

```

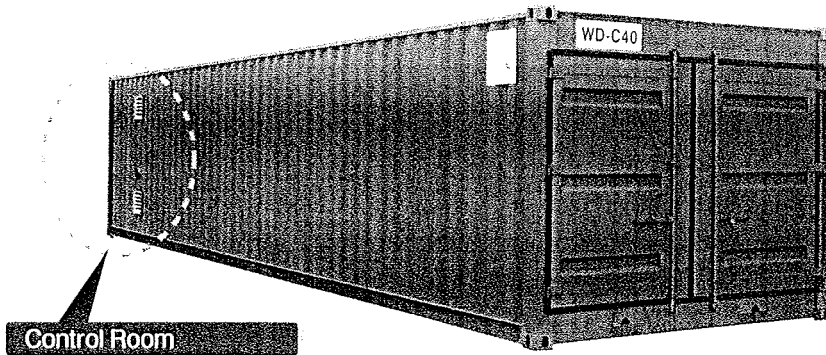
}
//-----

부 록 2.

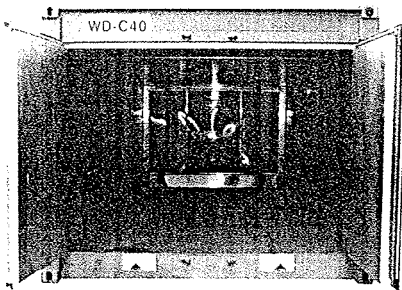
전자등 목재건조기

CATALOGUE

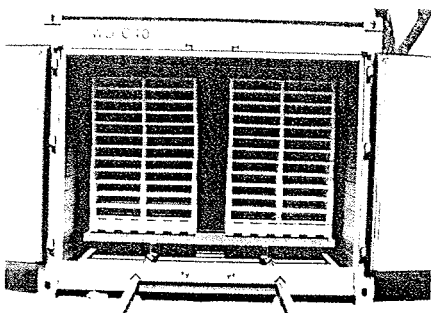
Automated Lumber Drying System (Model WD-C40)



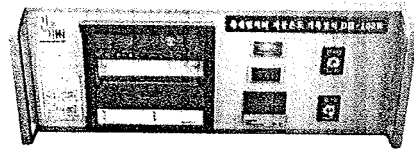
Control Room



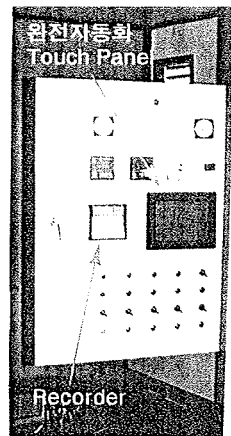
Drying Chamber



Heat-Treatment Of Pallet



Automatic Recorder For Heat-Treatment



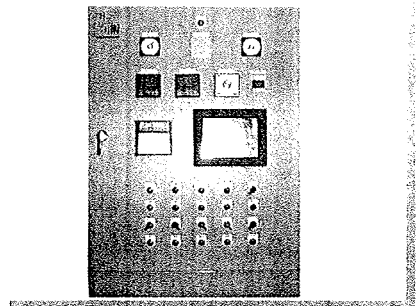
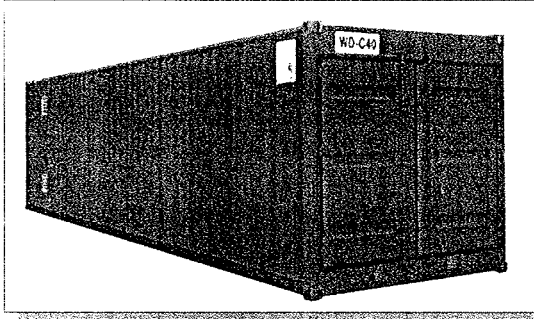
Control Room & Control Panel

주식회사 건조기술
術 Drying Engineering, Inc.

광주광역시 북구 월출동 973-4
Weolchul-Dong 973-4, Buk-Gu Gwangju 500-460, Korea
TEL : (062) 973-7524
FAX : (062) 973-7527
www.dryingeng.com

Automated Lumber Drying System

Automated Lumber Drying System



Control panel

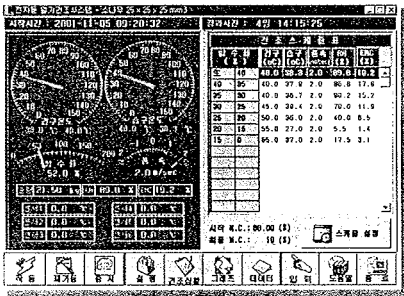
(touch screen type)

Specifications

- Model : WD-C40
- Heating Source : 75kW Elec. Heater
- Max. Temperature : 120°C
- Relative Humidity : 5~95%
- Workable Dimensions (W x H x L)
: 2,100 x 2,000 x 7,850 (33 m³)
- Total Weight : 5,000 kg

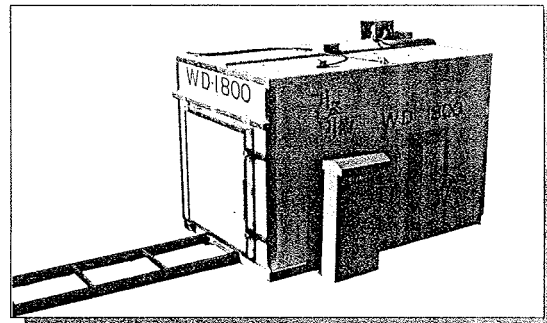
Automation System

- Sensors(Temp, RH, MC)
- PID controllers
- Computer-interface(RS485C)
- Lumber core temperature
- Software
- Automatic air speed control
- Reversible air flow (time interval)



Specifications

- Model : WD-0000
- Max. Temperature : 120°C
- Relative Humidity : 5~95%
- Workable Dimensions (W x H x L)
: Based on Order
- Material : 100mm-thick Urethane panel



규격은 주문에 따라 현장 사공이 가능합니다.

주식회사 건조기술
Drying Engineering, Inc.

광주광역시 북구 월출동 973-4
Weolchul-Dong 973-4, Buk-Gu Gwangju 500-460, Korea
TEL : (062) 973-7524
FAX : (062) 973-7527
www.dryingeng.com

주 의

1. 이 보고서는 농림부에서 시행한 농림기술개발사업의 연구보고서입니다.
2. 이 보고서 내용을 발표할 때에는 반드시 농림부에서 시행한 농림기술개발사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 아니됩니다.