□ 特輯 : μ-Processor應用

Programmable Controller(PC)의 Software 개요와 개발사례

李 昌 求·崔 洛 滿·趙 三 鉉
(韓國電子技術硏究所)

■ 차 례 ■

〈Part I : PC 소프트웨어 개요〉

1 서 론

PC의 초기 형태는 간단한 순차공정을 기계적인 캠·샤프트 (cam shaft) 와 드럼 (drum) 을 사용하여 반복적인 작업을 수행하는 것이었다.

PC를 순차제어기 (sequential controller 혹은 sequencer) 라고도 부르는 이유이다.

이후 계전기 (relay) 의 출현으로 전기적인 순차제어 방식이 채택되었으나 공정제어의 대상이나 제어범위를 확장시키지는 못하였다. 단지 무접점 계전기 (contactless relay) 의 개발이 다소 복잡한 공정을 고속으로 처리해 주기는 했으나 계전기 순차제어기가 가지는 근본적인 문제점들ㅡ 즉, 프로그램이 어렵고, 공정이 바뀔때 전체의 계전기제어반 (relay control panel) 을 폐기하고 새로운 제어반을 설계 제작하여야하는 불편한 점과 제어소자가 커서 설치 용적이 많이 소요되며 소비전력이 크다는 점등ㅡ 은 개선되지 않았다.

60년대 말 소개된 트랜지스타 및 IC가 도입됨으로써 소전력, 소용적, 고속프로그램 처리가 가능한 PC가 소개되어 계전기 제어반의 대치가 이루어져서 현장의 요구를 기술적으로는 어느정도 충족시켰으나 제품이 고가여서 보급활용단계에는 이르지 못하였다. 그러나 70년대 중반 이후부터 마이크로프로세서 (microprocessor) 의 PC응용이 본격적으로 전개되면서 상황은 급변하고 있다.

계전기 래더 (relay ladder) 에 대한 순차제어전용이던 종래의 PC개념은 복잡한 공정제어가 가능하고 필요한 공정정보의 수집, 저장, 처리, 표시및 문서관리 기능등을 다양하게 내장시켰으며, 당시까지 대형공장의 자동화에 도입되고있던 집중제어방식 (centralized control system) 은 점차 마이크로프로세서를 내장한 PC를 중심으로 한 분산제어방식 (distributed control system) 으로 급속히 전환하는 추세에 있다. 즉, 마이크로 프로세서 및 제어프로그램 (control program) 을 내장한 단위 PC들이 정보망 (data network) 을 통하여 상위 레벨의 컴퓨터 (host computer) 에 연결되고, 각 단위 PC는 각자가 맡은 공정을 제어하면서 필요한 정보를 상위 레벨의 컴퓨터와 주고 받는다. 자연히 PC 제조원 (PC makers) 들은 계층구조 (hierarchi-

cal structure) 를 갖는 제품들을 다양하게 갖추어 분산제어구조를 갖는 PC계 (PC family) 를 구성하는 개발의 경향을 보이고있다.

일반적으로 볼때 오늘날까지의 PC가 있기까지 산업공정의 발전과정이 매우 복잡하고 다원적이기 때문에 PC의 발전방향을 어느 한정된 범위에서 파악하려는 것은 매우 어려우며, 전통적인 계전기 래더식 공정이나 instrumentation 및 process control 의 이해와 컴퓨터를 이용한 제어이론에 대한 전반적인 것에 관해 고려하지 않으면 안된다. 또한 앞으로의 산업공정이 로보트를 많이 도입하는 자동화 공장의 등장과 공정의 변화에 손쉽게 대응할 수있는 FMS (flexible manufacturing system) 체계로 전환하리라는 전망으로 볼때 PC의 발전이 공정 변화에 급속한 템포로 영향을 끼칠것으로 전망된다.

본고에서는 주로 PC의 사용자 입장에서 고려해야 할 소프트웨어 (software) 구성에 대한 고찰을 해보고 그것들이 공정자동화에 대한 산업체의 요구에 어떻게 부응할 수 있는지를 개발사례를 통해검토해본다.

2 시스템 소프트웨어

시스템 소프트웨어라 함은 프로세서, 메모리, 입·출력 장치 (I / O devices) 등의 시스템 자원 (system resources) 을 효율적으로 운영하고, 응용프로그램을 용이하게 개발할 수 있는 환경을 제공하여 주는 소프트웨어를 의미하며, 일반적으로 시스템 핵심 (kernel), 시스템 서비스루틴 및 시스템 utility 등으로 구성되는 운영체제 (operating system) 또는 모니터 프로그램 (monitor program) 을 말한다. 이러한 운영체제는 일반 컴퓨터의 시스템에서는 그 용도에 따라 범용과 특수용도의 운영체제로 구분되어지며, 대형 PC에서는 제어분야에 적합한 특수용도의 운영체제 (control application oriented operating system) 를 도입하여 사용하고 있으나, 소형 PC에서는 소량의 프로그램 메모리와 같은 PC자체의 제약성과 일반 컴퓨터 시스템과의 구성적인 상이성으로 인하여 일반 컴퓨터 운영체제를 그대로 도입하여 사용할 수 없는 경우가 많다. 그러나, 근래에는 PC에서의 마이크로 프로세서 사용의 보편화로 응용제어프로그램 개발에 필요한 기본적인 기능만을 가지는 소규모의 운영체계 (예, micro - 51, iRMX, VRTX, 등) 가 발표되고, 이러한 운영체제를 ROM에 내장시켜 마이크로 프로세서와 함께 제공하므로서 PC응용프로그램의 개발자로 하여금 PC의 기본적이면서도 복잡한 운영체계 (또는 모니터 프로그램) 을 자체적으로 개발할 필요 없이 응용프로그램의 개발에 전념하게 할 수 있도록 하여 주므로서 전체 시스템 개발기간을 단축시켜주며, ROM에 내장된 운영체제를 사용하므로서 얻을 수 있는 소프트웨어의 이식성과 신뢰성 제고 및 시스템 실행속도를 단축시킬 수 있는 장점을 제공받을 수 있다.

3 프로그램 언어

PC에서 주로 사용하는 프로그램 언어는 다음과 같이 네가지 형태로 나눌 수 있다.

—계전기 래더도형 (Relay ladder diagram)
—Boolean mnemonic 언어
—기능 블럭 (Functional blocks) 언어
—컴퓨터식 (English statements) 언어

계전기 래더도형이나 Boolean 언어는 PC의 기본언어이고 기능 블럭과 컴퓨터식 언어는 고급언어 (

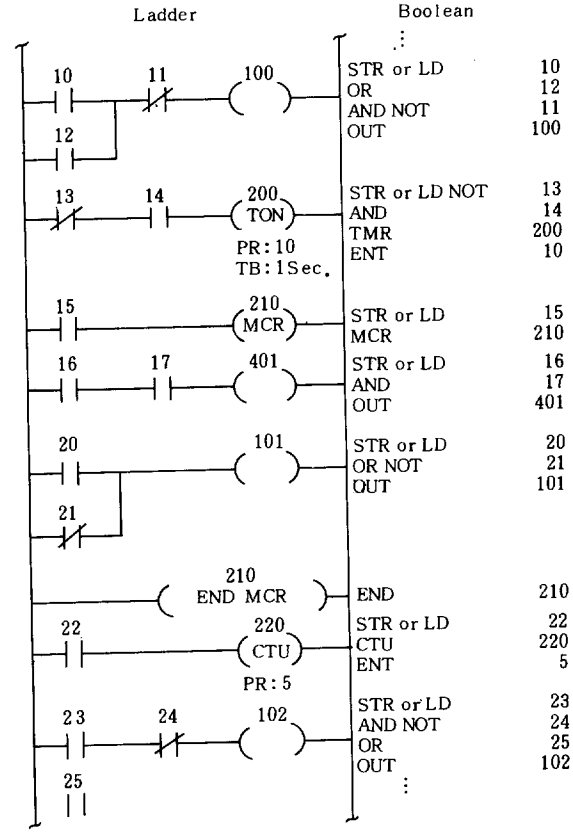


그림1. PC 래더도형과 등가Boolean 프로그램

high level language) 에 속한다. 기본 언어는 계전기식 순차제어에 필요한 순차제어, 계시 및 계수 (timing and counting) 제어와 논리회로 구성등을 프로그램하는데 주로 쓰이며, 고급언어는 아나로그 제어 (analog control), 데이타 처리 및 보고등기본언어로는 수행하기 힘든 복잡한 기능을 프로그램하는데 사용되고 있다.

초기의 PC는 계전기 대치가 주기능이었기 때문에 그림 1과 같은 래더도형이나 Boolean 언어로 프로그램하였다.

래더도형언어는 계전기의 코일이나 접점의 상태를 알기쉬운 기호로 표기하는 프로그램언어이며 비교적 초기 PC는 이 언어를 주로 사용하였으나 그후 마이크로 프로세서의 도입으로 데이타처리, 연산 및 프로그램 수행을 제어하는 언어구사가 가능해졌다. 기본 명령어로는 계전기 래더명령, 계시 및 계수지시

MNEMONIC	FUNCTION	LADDER EQUIVALENT
LD/STR	Load/Start	!-] [-
LD/STR NOT	Load/Start Not	!-]/[-
AND	And Point	-] [-
AND NOT	And Not Point	-]/[-
OR	Or Point	! / ! / -] [-! / !
OR NOT	Or Not Point	! / ! / -]/[-!
OUT	Energize Coil	-( )-
OUT NOT	De-Energize Coil	-(/)-
OUT CR	Energize Internal Coil	-( )-
OUT L	Latch Output Coil	-(L)-
OUT U	Unlatch Output Coil	-(U)-
TIM	Timer	-(TON)-
CNT	Up Counter	-(CTU)-
ADD	Addition	-(+)-
SUB	Subtraction	-(-)-
MUL	Multiplication	-(×)-
DIV	Division	-(÷)-
CMP	Compare=, <, >	-(CMP)-
JMP	Jump	-(JMP)-
MCR	Master Control Relay	-(MCR)-
END	End MCR, Jump, or Program	-(END)-
ENT	Enter Value for Register	Not Required

그림2. Boolean 명령어와 등가 래더기호

명령, 연산, 데이타처리, 프로그램제어등의 기능을 가진다. Boolean 언어는 기본적인 Boolean연산자인 AND, OR, NOT이 그 주요 구성요소이며 그림 2에서 볼 수 있듯이 몇개의 mnemonic 언어가 래더도형을 나타내기 위해 정의되어 있으며, 기본지시 명령어는 래더도형의 경우와 유사함을 알 수 있다.

기능블럭언어는 고급언어 (high level language) 의 일종으로 그림 3과 같이 특정기능을 블럭으로 표시하고 입출력 제어조건을 표기하는 방식이다. 전술한 두가지 기본 PC언어가 수행하는 기능 이외에 데이타 변환, 파라미터설정, 블럭데이타이동, ASCII 변환, 스택변환등의 기능이 추가된다. 특수한 경우 PID (proportional -integral - devivative)블럭이나 드럼계시기 (drum timer) 의 등가인 순차제어 블럭 (sequence block) 등의 기능도 가진다.

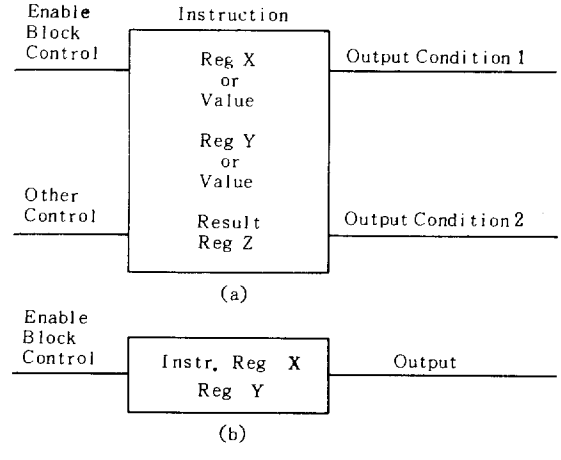


그림3. 기능 블럭언어 표시 예

컴퓨터식 PC 프로그램 언어는 그림 4와 같이 일반 컴퓨터가 사용하는 BASIC, ADA, PEARL과 같은 언어를 말하며, PC에 적용된 경우 조작자가 익히기에 더 간편하게 되어있다. 이 언어를 사용하면 PC의 입출력이 수치 데이타를 직접 레지스타 (register) 에 옮길 수 있고 주변기기와 정보교환이 용이할 뿐만 아니라 기능 블럭으로 처리할수 없는 업무도 수행할 수 있어 주로 대형 PC(입출력점이 1000~4000이상) 언어로 이용되고 있다. 그러나 PC 제작사마다 특성있게 개발된 명령어를 별도로 쓰고있고 명령어의 수도 PC의 규모에 따라 다르기때문에 PC선정시 면밀히 검토하여야 할 사항이다. 이상 개략한 네가지 언어는 그사용 레벨에 따라서 각각 장·단점을 가지고 있고, PC의 경우 일반컴퓨터와는

```
10  REM 12 CHANNEL PID LOOP CONTROL
20  DIM RESET (12), UPDATE_TIME (12),
        INTEGRAL (12), DERIVATIVE (12),
        PROP_BAND (12), RATE (12)
30  FOR C=0 TO 11: REM C=CHANNEL
40    PROCESS_VARIABLE=AIN (C)
50    AOT (C) =PID_LOOP (C)
60  NEXT

REALFUNCTION:PID_LOOP
10  EXTERNAL RESET, INTEGRAL, DERIVATIVE,
    UPDATE_TIME, PROP_BAND, SETPOINT,
    RATE, PROCESS_VARIABLE
20  INTEGERARGD
30  PROCESS_ERROR=PROCESS_VARIABLE-
    SETPOINT (D)
40  INTEGRAL (D) =RESET (D) * (INTEGRAL(D)
    +PROCESS_ERROR) * (UPDATE_TIME (D)
    -TIMER)
50  DERIVATIVE (D)=RATE (D) * (PROCESS_
    ERROR-DERIVATIVE (D) / (UPDATE_
    TIME (D) -TIMER)
60  CONTROL_OUTPUT=PROP_BAND (D) *
    PROCESS_ERROR+INTEGRAL (D) +
    DERIVATIVE (D)
```

그림4 고급언어로 쓰여진 PID제어 프로그램

달리 사용언어의 선택이 PC본체의 선택과 분리할 수 없으므로 언어의 선택은 제어대상 공정에 대한사전 충분한 지식과 PC의 내용을 면밀히 검토한 후 결정되어야 한다. 사용자 입장에서 프로그램 언어의 선택시 고려해야 할 사항은 다음과 같다.

— 사용의 편이성
— 언어의 기본특성
— 해결해야 할 문제점의 분석
— 프로그램 수행속도 등

이상 열거한 사항들은 현장 엔지니어들의 경험적인 인습에서 오는 편향성과 시스템 운영 및 유지보수를 담당할 인력의 자질들을 감안하여야 하며 추후 증설 운영 및 자동화계획과 연결하여 종합적인 결론을 내리는 것이 현명할 것이다.

4 PC 프로그래밍

PC의 프로그래밍을 통해 운용소프트웨어를 만드는 일은 매우 간단하다. 특히 PC의 입출력수가 적거나 기존의 계전기 대치용만으로 쓰일 경우는 거의 문제가 되지 않는다. 문제가 되는 것은 입출력의 수가 많고 복잡한 제어기능을 요구하는 공정에 대한 프로그램을 만드는 일이고, 그것은 PC 그 자체의 설계만큼이나 복잡하고 조직적인 접근을 요구한다. 즉, 엔지니어가 머리속으로 구상한것을 가지고 바로 키보드 앞에 앉아 프로그램하면 되는 것으로 생각하면 십중팔구는 일을 그르치고 말 것이다.

공정자동화를 위한 전체적인 제어구도를 대상으로 요구사양에 대한 적절한 도면이나 개발도구를 이용하여 시스템을 설계하고, 이것을 근거로 구체적인 논리도 (logic chart) 를 작성한다. 이상을 토대로 코딩 (coding) 을 하여 오프라인 (off -line)으로 코드를 점검한 후 최종적으로 시스템 종합과 점검에 들어간다.

소프트웨어 설계과정은 많은 전문인력과 시간이 소요되나, 설계단계에서의 정확성은 프로그램 코딩, 테스트, 및 유지·보수등의 시스템개발 전 과정을 통하여, 지대한 영향을 주며, 시스템 개발· 운영비의 결정에 많은 비중을 차지하고 있으므로 정확한 시스템 설계에 많은 노력이 요구되어진다.

소프트웨어의 개발에 필요한 상기한 요구들이 인적자원과 경험이 부족한 산업환경에서 PC의 보급이 원활히 되지 못하게 하는 요인으로 작용하고 있어 PC 제조회사들은 초기 설치 및 시운전에 대한 지원계획을 가지고 임하고 있으나, 현장 엔지니어가 PC 소프트웨어에 대해 숙지하고 있지 않는한 공정변경이나 증설시 문제에 봉착하게될 것이다. 따라서, 산업체 현장 엔지니어들은 그들이 사용하는 PC의 운용 소프트웨어를 충분히 분석·이해하고 추후의 기능추가 및 새로운 시스템을 개발할 수 있는 능력을 갖추는 것은 매우 중요하다. 여기서는 시스템 개발을 위한 소프트웨어 개발 과정을 고찰하고자 한다.

4.1 자료수집 및 점검

제어대상공정의 각종 기기 및 설치에 관한 공정도면, 계장설치도면, 제어반도면, 동력제어실 설계도면등과 PC의 입출력에 대한 기술자료를 수집하여 분류한다. PC 입·출력단자에 연결할 각종 스윗치, 계전기, 경보장치등의 정상상태를 규정하고 해당 입·출력을 정의한다. 그리고, 공정이 격게 될 모든 동작 조항이 포함되는 시스템작동시나리오 (system operation scenario) 를 면밀히 작성하고 반복하여 점검한다. 시나리오는 특히 모든 하드웨어 (hard - ware) 적인 고장요인에 대해 PC의 출력단이 구동되지 않은 상태 (deenergized state) 에서 안전한지에 대해 반복적으로 확인점검되어야 한다.

4.2 공정의 기능사양서

소프트웨어 개발에 필요한 사양서를 공정자료, 시스템동작등을 고려하여 간결한 형태로 표현한 기록을 총칭하는 것으로서 코딩을 하기위한 자료로서 활용할 수 있는 입·출력명단 (I/O list), 입출력 번지배정 (I/O address assignment), 흐름도 (flow chart), 논리도 (logic diagram) 등이 있고 일반적인 시스템의 기술사양과 하드웨어 및 소프트웨어 전반에 대해 정의하여야 한다. 소프트웨어 기능사양은 공정제어 기능을 유효한 기능블럭 (functional block)으로 나누고, 각 기능블럭을 코딩할 수 있는 정도의 세부모듈로 구성하여 기술한다. 각 모듈은 입출력 상황의 여러 다른 조건하에 어떤 기능을 가지는가에 대해 기술되며, 기술된 내용이 복잡해지지 않도록 써브모듈 (submodule)로 더 분류 할수도 있다. 기능사양서는 되도록이면 간략하게 표시되어야 하며 사양서 작성자와 프로그래머가 다른 경우에도 해석의 차이가 없도록 명확하게 정의 되어야 한다.

4.3 코딩

코딩은 전술한 논리도 (logic chart) 나 계전기도 형등으로 부터 PC 래더프로그램을 작성하는 과정을 의미하며, 코딩된 래더프로그램은 그림 5와 같이 대상 기계나 공정을 직접적으로 제어하는 형태로 표시되며, PC내의 사용자 프로그램영역 (user application program memory area) 에 저장된다.

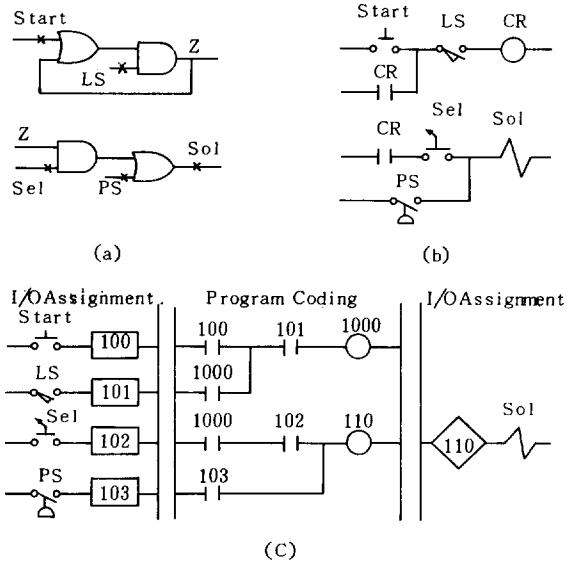


그림5. PC 프로그램 코딩 예

이상의 작업으로 일단 코딩이 끝나면 에뮤레이션을 통해 하드웨어 및 소프트웨어를 시스템차원에서 tuning 하고 테스트하며, 확정된 프로그램은 PC의 소프트웨어 패키지화 하고 PC에 내장시켜 현장시험에 들어가게된다.

5 PC의 내일

PC의 내일을 예측하는데 있어 고려해야할 두가지 관점이 있다. 그 하나는 기술의 진보에 대한 예측이고, 다른 하나는 내일의 산업 현장에 대한 적용성에 관한 것이다. 아마도 후자의 요구도 여하가 전자의 향방을 가름할 것이다. PC는 재고관리, 유지보수진단, 보고서 작성등과 같은 더 복잡한 고기능을 보유하여 산업 환경하에서 미니컴퓨터와 효과적인 경합을 벌일 수 있게 될 것이다. 지금의 일부 PC 시스템은 이미 텔레타이프나 라인프린터 및 CRT 터미날과 같은 컴퓨터 주변장치를 부착하여 사용하는 시스템도 상품화되어 있다. 현재 시도되고 있는 PC는 PC간 데이타망 (inter-controller communications links), PC와 host 간 데이타 고속도로 (data highway) 망은 분산제어시스템의 급격한 진전을 예상하게 하며, 한편으로는 각종 정보처리에 필요한 데이타의 저장·검색을 용이하게 하기 위한 데이타 베이스 시스템, PC 자체의 고장으로 인한산업재해 방지를 위한 시스템의 고장인내 (fault tolerant) 시스템화, 공정정보를 정확하고 쉽게 오퍼레이타에게 전달하기 위한 그래픽 (graphics)의 도입, 및 PC의 확산·보급에 중대한 요소인 user-friendly 시스템화하는 경향이 있다. 이러한 기술의 추세로 미루어 볼때 내일의 PC의 적용은 현재의 PC와 마찬가지로 생산성재고와 비용절감의 측면에서 주로 고려가 될 것이나, PC는 새로운 제품의 개발 뿐만 아니고 새로운 산업체의 등장에도 기여하게 될 것이다.

〈Part II : 개발사례(자동온도 제어기)〉

1 시스템 개요

컴퓨터의 발달과 자동제어 이론의 발달은 공장자동화 분야의 혁신을 가능케 하였으나, 실제 적용과정에서는 기존 산업구조와의 마찰로 상당한 어려움이 있었다. 그러나 도입 시스템의 성공적인 운영과 computer mind 의 확산, 마이크로프로세서의 기능

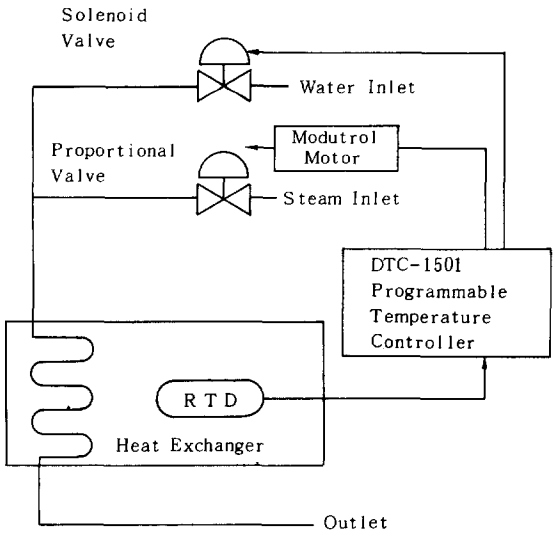


그림6. 염색기 온도제어시스템 구성

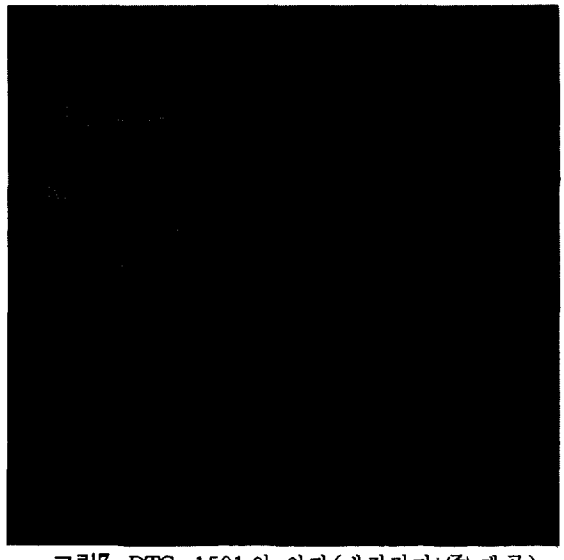


그림7 DTC-1501의 외관(대정전자㈜ 제공)

향상에 따른 응용범위의 확대, 계속적인 하드웨어가격의 하락등으로 digital control 은 공장자동화를 위한 작은시스템으로부터 큰시스템에 이르기까지 산업제어의 전 분야에 걸쳐 그 응용 영역을 확대해 나가고 있으며 특히 최근에는 digital instrumentation분야에서도 성장을 계속하고 있다. 그러나 국내여건은 대부분의 중소기업의 경우 재래적인 방법을 답습하고 있으며, 대규모 플랜트의 경우에는 직접 turn - key base 로 도입 운영하고 있어 PC의 자체개발기반이 확립되어 있지 않는 실정이다. 이러한 차제에 전자기술연구소는 중소기업에 대한 기술지도 계기를 마련하여 공정제어중에서도 가장 광범위하게 요구되는 온도제어를 위한 PC를 개발하기에 이르렀다. 이번에 개발한 DTC-1501은 특히 염색기의 내부온도제어를 위하여 설계된 것이나, 일반적인 산업의 온도제어공정에도 적용할 수 있다. 이 시스템은 온도감지기 (RTD sensor) 로 부터 데이타를 받아 염색기 내부온도가 프로그램된 온도패턴 (pattern)을 유지할 수 있도록 그림 6과 같이 스팀밸브와 냉각수밸브를 구동시키는 시스템이다. 현장 방문을 통한 환경조건 및 요구사항을 분석한 결과를 토대로 시스템 설계에 임했으며 초도품이 그림 7과 같이 프로그램 키보드 (key board), LED display, 경보램프등으로 조작자 (operator) 와 인터페이스 (interface) 되고 있다.

본장에서는 시스템 전체의 설명은 피하고 소프트웨어 개발을 중심으로 논하기로 한다. 소프트웨어 부분은 크게 monitor 프로그램과 제어프로그램 (control program) 으로 구분하여 개발하였다. monitor 프로그램은 시스템초기화 (system initialization) 와 입출력제어, key service routine 등으로 구성하였으며, 제어프로그램은 비례밸브제어를 위한 PI제어프로그램과 솔레노이드밸브제어를 위한 on-off제어 및 부동소수점연산 (floating point arithmetic)으로 구성하였다.

2 제어 앨고리즘

열교환기에 적용된 digital algorithm의 특성과 장단점에 대한 비교 결과는 이러한 algorithm들이 재래적인 연속제어계의 PID algorithm 공법보다 더 좋은 결과를 보여 주지는 못하였다. 본 project에서는 재래적인 연속제어계의 PI algorithm을 기본으로 Dahlin's algorithm에 따라 설계된 digital PI algorithm과 비교하여 비례계수 K_P, 리셋타임 T_i 를 유도하였다.

일반적으로 연속제어계에 있어서 PI algorithm은 다음과 같은 두가지로 표시한다.

$$D(s)=K_P\left(1+\frac{1}{T_iS}\right) \qquad (1)$$

$$D(s)=K_P+\left(1+\frac{1}{T_iS}\right) \qquad (2)$$

여기서 (1) 식은 계수 K_P 와 T_i 의 연관성을 부여하는 것이고, (2) 식은 K_P 와 T 를 독립적으로 해석하는 것이다. 본 project 에서는 (1) 식을 사용하였으며 이를 불연속 제어계로 표시하면 아래

식과 같다.

$$D(Z)=K_P(1+\frac{T/T_i}{1-Z^{-1}}) \quad (3)$$

PI algorithm을 적용하여 공정을 제어할 경우 비례계수 K_P와 리셋타임 T_i를 결정하는 것이 큰 문제가 되며 여기서는 Dahlin's algorithm에 의해 설계된 제어계의 특성식 D(Z) 로 부터 tuning 계수 K_P, T_i를 유도하였다.

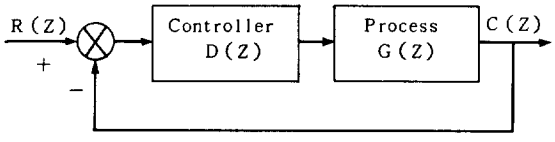


그림8. 제어계 블럭도

그림 8에서 $C(Z)/R(Z)=\frac{D(Z)\cdot G(Z)}{1+D(Z)\cdot G(Z)}$ (4)

가 되고, 따라서

$$D(Z)=\frac{1}{G(Z)}\cdot\frac{C(Z)/R(Z)}{1-C(Z)/R(Z)} \quad (5)$$

가 되므로, 만약 G(Z)가 주어진다면 입력 R(Z) 에 따라 원하는 출력 C(Z)를 얻도록 제어기 D(Z)를 설계 할 수 있다. 여기서는 G(Z)가 1차 지연함수로 표시되고, unit step input R(Z)에 대한 출력 C(Z)가 바람직한 형태, 즉 ITAE = 0.575, IAE = 0.758을 갖도록 제어기 D(Z)를 설계하였으며 이 과정에서 D(Z)는 식(6)과 같이 표시된다.

$$D(Z)=\frac{1-e^{-T/\tau_f}}{Kg(e^{T/\tau}-1)[1+N(1-e^{-T/\tau_f})]}\cdot(1+\frac{e^{T/\tau}-1}{1-Z^{-1}}) \quad (6)$$

이 식을 (3)식과 비교하면

$$K_P=\frac{1-e^{-T/\tau_f}}{Kg(e^{T/\tau}-1)[1+N(1-e^{-T/\tau_f})]} \quad (7)$$

$$T_i=\frac{T}{e^{T/\tau}-1} \quad (8)$$

과 같이 된다. 여기서

T : Sampling Time

τ : Process Time Constant

Kg : Process Gain

τ_f : Design Time Constant

τ_d : Process Delay Time 이며 $N=T/\tau_d$ 이다.

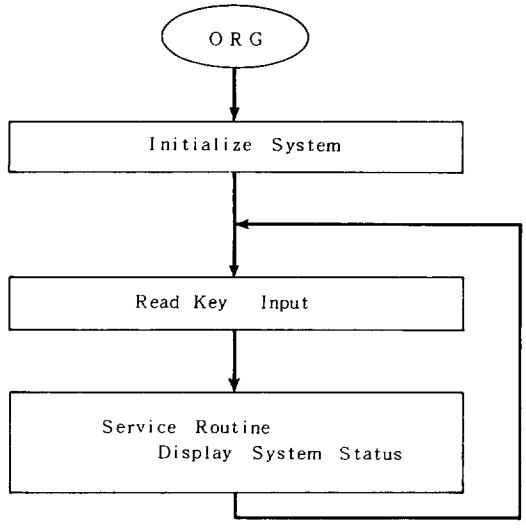
③ 소프트웨어 개발

3.1 Monitor 프로그램

시스템을 초기화(initialization) 하는 부분과 사용자의 제어 프로그램을 읽어드리고 시스템의 상태를 7-segment LED에 표시하는 입출력제어부분, 그리고 각 key 에 대한 service routine 등으로 구성되어있다.

1) Monitor 흐름도

Monitor 프로그램은 약 2K byte 의 크기를 가지며 전체적인 흐름도는 그림 9와 같다.



- System Parameter
- RTI A/D Converter
- 8255 I/O Port
- 7-seg LED & Lamp

그림9. Monitor 흐름도

2) 상태천이도(State transition diagram)

Monitor 프로그램의 설계를 체계적으로 하기 위하여 그림 10과 같은 상태천이도를 작성한다. 그림에서 이중타원은 function key 가 입력되었을경우 정해지는 상태를 표시하며 나머지 타원은 각 상태에 대한 minor 상태 또는 상태변화중간의 intermediate minor 상태를 표시한다.

3) Buffer 와 Table

시스템의 상태를 표시하기 위하여 여러개의 buffer 및 table 들을 정의하였으며, 이중 주요한 것은 다음과 같다.

• INPRQC : In-Process Flag로 process 의 진

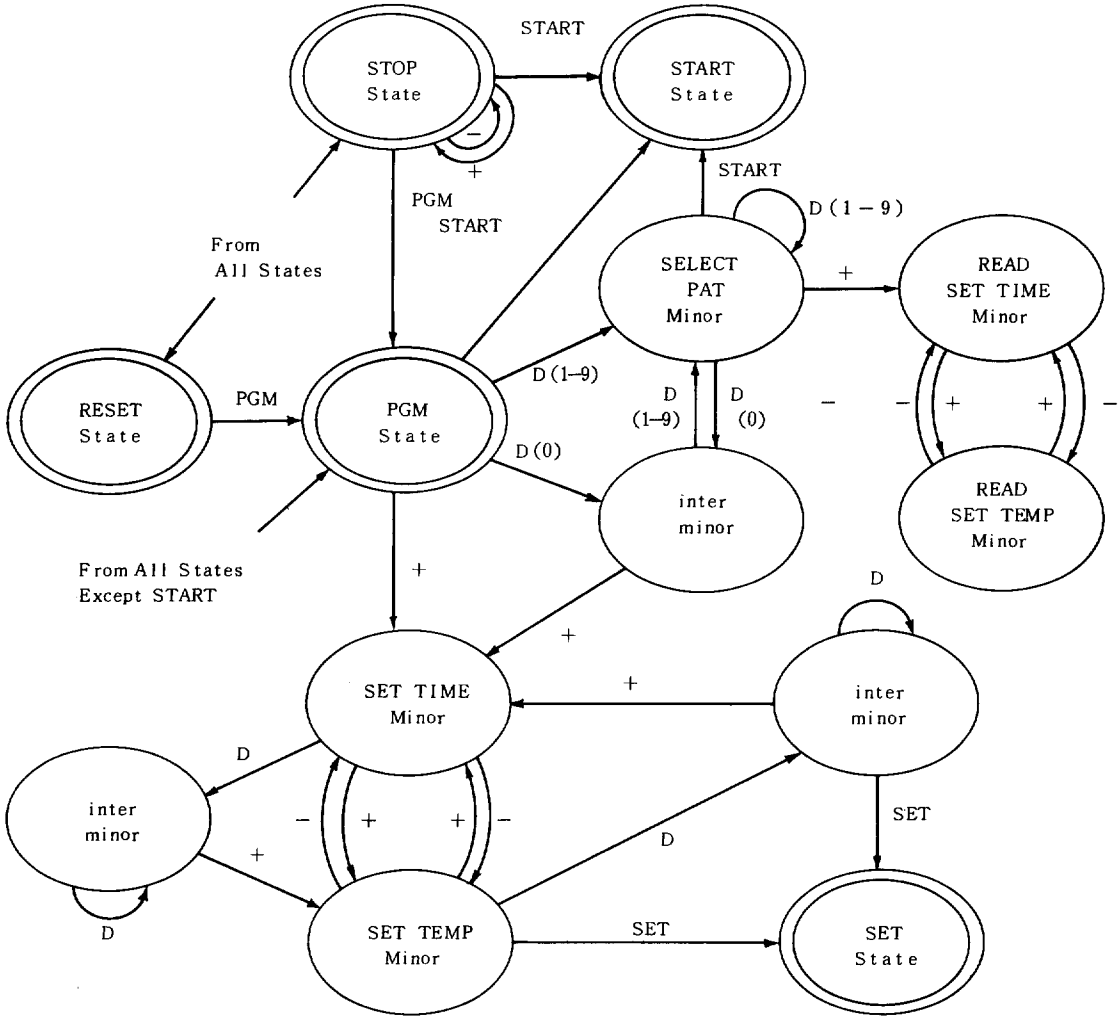


그림 10. DTC-1501 상태천이도

행 상태를 표시한다.
START key 입력시 bit 0를 set
STOP key 입력시 bit 7을 set

- STATE : Function State Buffer 로 function key 입력시 해당 state (0 – 4) 가 load 된다.
- DISBUF : Display Buffer 로 각 7-segment 에 display 된 image pattern을 가지고 있으며 8 byte 로 되어있다.
- MINOP : Minor State Buffer 로 현재의 minor state 를 보관한다.
- KEYTAB : Key Code Transform Table로 position key code 를 internal key code 로 변환시킨다.
- TITAB : Time Table 로 user control program 중의 time value 를보관한다.
- TETAB : Temperature Table 로 user control program중의 temperature value 를 보관한다.
- SLOP : Slope Table 로 user program으로부터 시간에 대한 온도변화율을 계산하여 floating point format 으로 보관한다.
- DISTAB : Display Pattern Table 로 IMPAT, ERRTAB, DASH등이 이에 속한다.

4) Subroutines

㉮ 시스템 초기화 (System Initialization)

- 8255 Initialization

2개의 8255 I/O controller 를 mode 0로 set 시키고 각각의 port 를 목적에 맞게 input 혹은 output port 로 정의한다.

- RTI Initialization

  A/D converter 로 쓰이는 RTI 의 gain selection, MUX address, pacer selection 을 정의한다.

- Buffer Initialization

  system 에서 쓰이는 여러가지의 buffer를 초기화 시킨다.

- Display Device Initialization

  " -PC UP " 을 display 하고, 모든 lamp 를 turn-off 시킨다.

㉮ Monitor Main Stem

Monitor 프로그램의 stem은 key 를 읽어드리는 READKEY, 읽어들인 key 를 해석해서 service 할 routine 을 찾아내는 KEYEXEC, 각 key 에 대해 실제의 명령을 수행하는 subroutine 들과 이때 필요한 여러개의 subroutine 들로 되어있다.

- DSPLY

  Display buffer (DISBUF) 에 있는 image pattern 을 7-segment LED 에 내보내어 display 시키는 routine 이다.

- READKEY

  Key input 을 읽어들이는 routine 으로 다음에 설명할 SCAN이라는 routine 을 call 해서 key 를 읽어 들인다. 일단 이 routine 에 control 이 넘겨지면 control 은 key 가 눌려질때 까지 이 routine 에서 빠져나가지 못한다. key 입력시 발생하는 debounce 문제는 이 routine 에서 해결한다.

- KEYEXEC

  이 routine 에서는 입력된 key 들로 부터 state 혹은 minor state 를 정의하고, 입력될 decimal key 를 적당한 buffer 에 load 시키는 routine 으로 control 을 넘겨준다.

- SCAN

  이 routine 은 3×6로 구성된 key matrix를 8255를 통해 차례로 읽어, key 입력 여부를 결정하며 입력된 key 의 position code를 buffer 에 load 한다.

- RPGM

  TITAB 과 TETAB에 있는 data 를 BCD form으로 바꿔 INBUF에 옮기고 다시 이것을 display pattern 으로 바꿔서 DISBUF에 load 한 다음 RETURN한다. 이 routine 에서는 INTAB, DISIN, PATPT 등의 routine 을 CALL 하며 PGM key 가 눌러졌을때 수행된다.

- NKEY

  Numeric key 가 입력되었을 때 수행되는 routine 이며 PGM의 각 mode 와 minor 상태에 따라 적당한 buffer 에 저장하고, DISBUF 에 저장한다.

- RINC

  INC key 가 눌러졌을때 수행되는 routine 이며 minor state 를 하나씩 증가 시킨다. PGM state 와 STOP state 에 따라 수행되는 routine 이 달라진다.

- RDEC

  DEC key 에 대한 service routine 이며 minor state 가 하나씩 감소한다.

- RSET

  TITAB 의 time 이 단조 증가하는가를 검토하고, table 의 마지막에 EOF (End of File) 을 marking 한다.

- RSTART

  이 routine 은 START key 가 눌러진후 수행되며 time buffer를 0으로 초기화 시키고, 현재 온도를 읽어 들여 TMP 7-segment 에 display 한 후 A/D converter 로 동작하는 RTI 로 부터 interrupt 를 기다린다.

  이 routine에 control 이 넘겨지면 STOP key 만 SCAN하고 다른 key 는 무시된다.

- PSTOP

  STOP key 가 눌러지면 공정의 control 을 중지하고 PSTP TIME이 count 되며 RUN TIME 과 SET TMP 가 display 되고 program을 verify 하기 위해 다른 key 가 입력되기를 기다린다. Control 은 main stem으로 넘겨진다.

3.2 제어프로그램

1) 온도제어프로그램

온도제어는 START key 가 눌러졌을 경우 먼저 현재온도가 설정온도에 어느정도 근접하여 있는가를 점검하여 정해진 한정값 안에 들어오도록 만든 후 real-time routine 으로 들어가도록 하였다. Real-time routine 에서는 on-off 제어와 PI 제어 algorithm을 조합하여 valve 에 가해지는 시간과 방향을 결정하여 actuator 를 동작시킨다. 온도제어에 대한 흐름도는 그림 11과 같다.

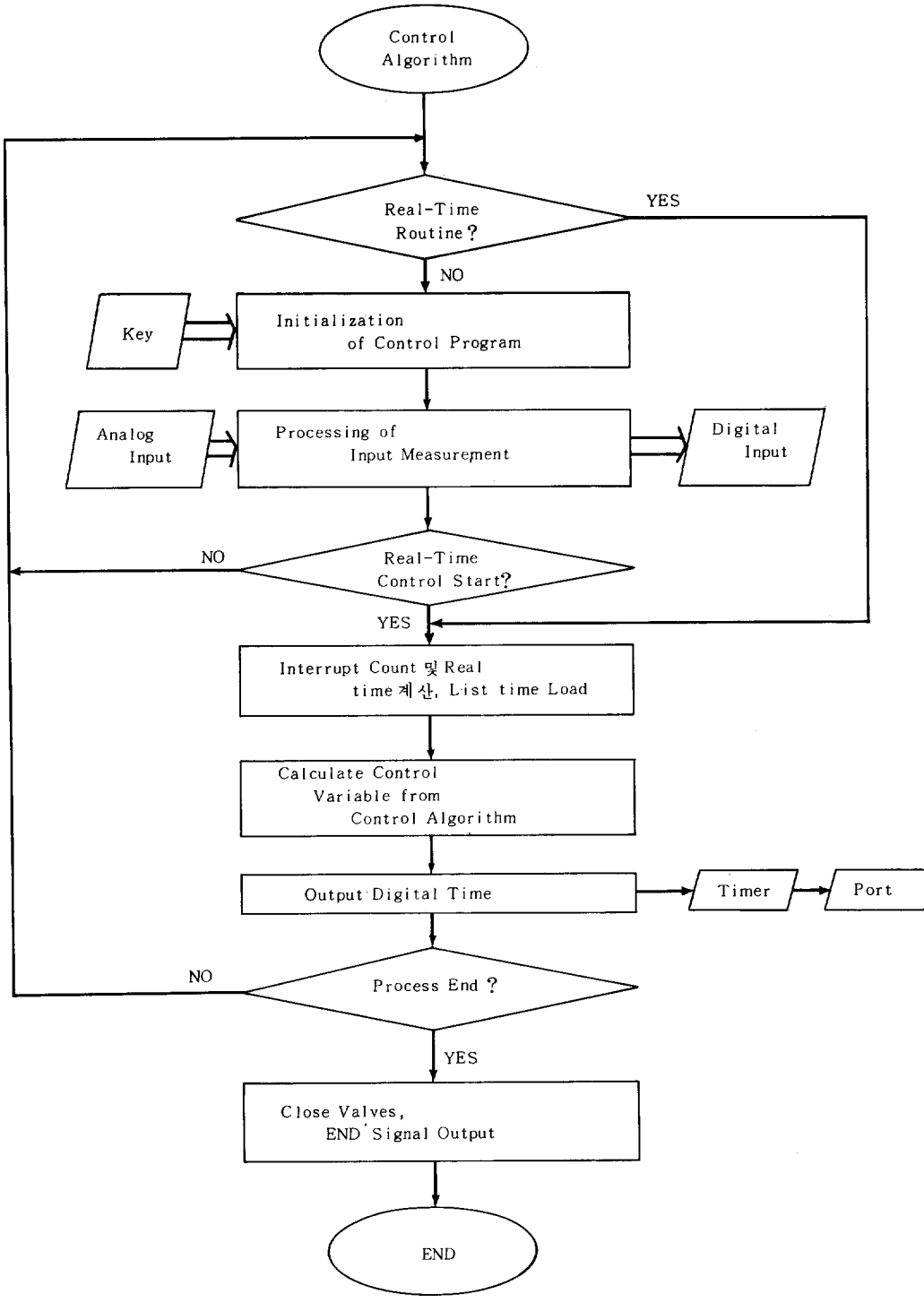


그림 11. 온도제어의 흐름도

2) Floating Point Arithmetic

요구되는 수치데이타의 값이 너무 크거나 작을때 한정된 byte 로서 좀 더 높은 정밀치를 얻기 위해 floating-point arithmatic 을 사용하여 다음과 같이 표시한다.

$$N = M * 2^{E} \quad (9)$$

여기서 M은 mantissa 로서 2 byte 로 표시하였고, E 는 exponent 로서 1 byte 로 표시하였다. 이런 format 으로 표시될 수 있는 수의 영역은 그림 12와 같다.

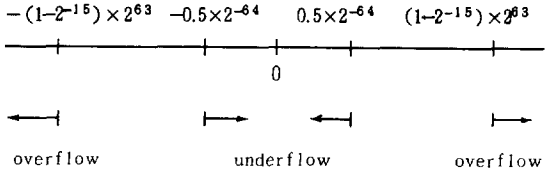


그림 12. 식(9) 로 표시되는 수의 영역

Floating-point arithmetic 을 위하여 다음과 같은 subroutine 들을 개발하였다.

- FIXFLO : Fixed-point to Floating-point
- FLOFIX : Floating-point to Fixed-point
- SUBFLO : Floating-point Subtraction/Addition
- MULFLO : Floating-point Multiplication
- DIVFLO : Floating-point Division

3.3 기타 프로그램

전술한 사항 이외에 온도쎈서의 비선형특성을 보상하는 프로그램과 현장의 잡음 (noise) 에 대한 소프트웨어적인 처리등이 주요한 개발내용이나 상당한 volume 이어서 본란에서는 생략하기로 한다.

4 맺 음

DTC-1501 은 아나로그 입력을 갖는 온도제어 전용 PC 로서 시스템개발의 전 공정을 국내에서 시도하여 상품화에 성공시켰다는 데 큰 의의가 있다.

본 project 는 대정전자(주) 에서 하드웨어 설계를 담당하고, 당 연구소가 소프트웨어 개발 지원을 하는 형태로 진행되었으므로 산·연협동연구 및 중소기업 기술지원의 성공적인 사례로 기억되기를 바라고싶다.

참고문헌

1) C.T.Jones, L.A.Bryan ; Programmable Controllers, IPC/ASTEC, 1983

2) Russell M. Loomis ; PC's Boost Quality Management Control and Productivity, IE September 1982

3) James Andrew Rovnak, Wayne C. Dunlap, Heniz B. Opladen, and James A. Mann : The Impact of Microprocessors on Process Control, COMPUTER DESIGN/April 21, 1983

4) Analog Devices ; Analog dialogue, Programmable M MAC 5000, Volume 17, Number 3, 1983

5) 전자기술 연구소 ; 염색기 자동화 사업 최종 보고서, 1984