

프로그래머블 콘트롤러의 개발 (A Study on Development of a Programmable Controller)

金龍洙, 金榮憲, 崔榮奎, 卞增男*

(Yong Soo Kim, Yeong Heon Kim, Young Kiu Choi and Zeungnam Bien)

要 約

본 논문은 일반적인 제조공정에서 시퀀스 제어용으로 사용될 수 있는 프로그래머블 콘트롤러의 설계 제작에 관한 것이다.

본 PC는 Z-80A 마이크로프로세서를 근간으로 하여 하드웨어로는 프로그래밍장치, 입력/출력 모듈, 타이머/카운터 모듈과 정전 복구기능 모듈이 제작되었고, 소프트웨어로는 초기상태화 프로그램, 모니터 프로그램, 실행 프로그램 및 정전복구 프로그램이 개발되었다. 특히 한 개의 타이머를 각각 다른 시간 간격으로 여러번 사용할 수 있게 하는 기능과 스캔타임을 줄일 수 있도록 실행 프로그램에 스킵 기능을 새로이 개발하였다.

Abstract

A programmable controller (PC) which can control general sequential process is developed. The Z-80A microprocessor-based PC includes hardwares such as programming device, input/output modules, timer/counter modules, and power-failure recovery module which for software, initialization program, monitor program, execution program, and power-failure recovery program are developed. In particular, the PC is designed in such a way that a timer can be used several times in different time intervals and a skip capability is incorporated in the user program to reduce scan time.

I. 序 論

Programmable controller (PC라 약칭함)가 개발 초기에는 시퀀스 제어를 위해 사용된 릴레이 회로에 공정변화에 따른 변경가능성(flexibility)을 주기위해 개발되었으나,^[1,2] 마이크로프로세서의 발달과 더불어 수행속도가 빨라지고 수행기능이 다양화되어, 아나로그 시그널을 취급할 수 있게 되어 피드백 제어(feedback control)도 할 수 있게 되었고, 최근에는 PID (proportional-integral-differential) 콘트롤 기능뿐만 아니라,^[2] 리모트 I/O 기술을 도입하여 다수의 PC

와 주(主)컴퓨터(main computer)와의 인터페이스를 통한 큰 규모의 플랜트 콘트롤로까지 확장되었다.^[4,5] 또한 독일의 포크스 바겐(Volks Wagen) 회사에서는 PC를 프로그램 가능한 고장진단 기구로써 활용하고 있다.^[6]

PC의 능력을 평가하자면 입력/출력 점접수 및 종류, 유저 프로그램(user program) 영역의 크기, 수행기능, 스캔 타임으로 비교할 수 있는 바, 외국에서 생산하는 PC는 소형에서 대형까지 종류가 다양하므로 종합적으로 비교 검토하기가 어렵다. 몇 가지 예를 들면, 일본의 Mitsubishi Electric의 MELM-100은 유저-프로그램 영역이 3,000byte이고, I/O 점접수는 1,024개이며, 수행기능에 있어서는 비교 기능, 쉬프트 기능등을 포함하고 있다. 또 Ohm electric의 PIMCO OR-32는 유저-프로그램 영역이 1,024bytes이고, I/O

*正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electronical Science, KAIST)

接受日字: 1983年 6月 23日

접점수는 128개이고, 수행기능은 상기한 PC와 비슷하다. 서독의 경우, Elon Scholtelemente의 DICAMP 4000은 유저-프로그램 영역은 256words(본 논문에서 다룬 PC는 약 1Kwords에 해당함)이고, 접점수는 126개이고, 역시 수행기능은 비슷하다. 영국의 경우, Allen-Bradley의 Bulletin 1750은 유저-프로그램 영역은 1,152words이고, I/O접점수는 186개이고, 수행기능은 비슷하다. 또한 스캔 타임에 관여하여 따로 비교해 보면 PMC-06은 40ms/1Kw이고, PLS-2형은 35ms/1Kbyte이다. 유저-프로그램 영역면에서 볼 때 PICOM OX-7은 1Kwords이고, satellite 100은, 1,024words이고, automate-3il은 2Kw이다.

이러한 외국의 발전 추세에 비추어 국내에서는 극소수의 상품이 소개되고 있을 뿐 아직 구체적인 PC의 개발 기술이 공개되어 있지 않은 실정으로서, 생산성 향상 및 자동화가 시급한 국내 여건을 고려하면 PC의 개발 및 일반에의 공개가 필요한 것으로 생각된다. 본 논문은 고급 고성능화 해가는 PC의 설계기술을 개발하고 발전시키기 위해 시도된 연구의 첫 단계의 결과로서, 그 내용은

- 1) PC의 사용자를 위한 프로그래밍 장치
- 2) 제어대상과 인터페이스하는 입력/출력 모듈과 타이머 모듈 및 카운터 모듈
- 3) 정전 복구기능을 가지는 모듈
- 4) PC에 내장되는 소프트웨어

이다. 특히 소프트웨어에 있어서는 스캔 타임을 줄이는 방법의 하나로서 스킵 기능을 개발하였으며, 타이머 사용에 있어서 새로운 기능을 개발하였다. 상기한 여러 외국의 PC와 비교하여 본 연구에서 설계 제작된 PC는 스캔 타임과 유저-메모리 영역에 있어서는 중형에 가깝고 수행기능에 있어서는 소형에 가깝다.

전체적으로 본 논문에서는 II장에서 시스템 구성에 관하여 설명하며, III장에서는 하드웨어에 관하여, IV장에서 소프트웨어에 관하여 설명하고 V장에서 실험 결과 및 검토, VI장에서 결론을 논하였다.

II. 시스템의 구성

PC는 여러가지 형태의 기계나 제조공정을 시퀀스 제어하기 위해 논리기능, 타이머기능, 카운터 기능, 연산기능을 수행하기 위한 인스트럭션들을 내장하고, 프로그래밍 장치를 사용하여 짠 user-program(ladder-diagram logic에 의해 구성)에 의해 입력/출력 모듈(input/output module)을 통해서 제어대상을 시퀀스 제어하는 컨트롤러로서,^[1,10] 본 연구에서 설계 제작된 PC에 대한 구성도는 그림 1과 같으며, 구성요소인 시

퀀서와 프로그래밍 장치, 입력/출력 모듈 및 이의 보조기능을 갖는 타이머/카운터 모듈에 대한 자세한 내용은 아래와 같다.

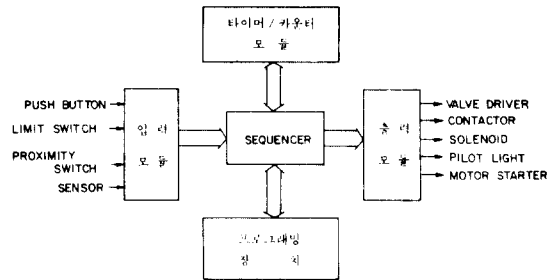


그림 1. PC의 구성도
Fig. 1. Block diagram of PC.

1) 시퀀서

Z-80A 4MHz CPU를 사용한 Cromemco 회사의 싱글 카드 컴퓨터(SCC)를 사용하여 구성되었으며, PC의 동작을 관장하는 부분이다. 즉 입력 모듈을 통해 들어온 정보를 사용자가 짠 프로그램에 따라 처리하여, 출력모듈을 통해 결과를 내보낸다.

시퀀서의 주요부분은 제어부와 PC를 초기화시키는 초기화 프로그램과 프로그래밍장치와 시퀀스 사이의 정보교환을 원활하게 해주는 모니터 프로그램과 제어대상을 유저-프로그램에 의해 시퀀스 제어하는 실행 프로그램을 내장하는 ROM과 유저-프로그램과 유저-프로그램 수행중 발생한 상태를 저장하는 RAM과 정전후 전원이 복구될 때 시스템을 정전전의 상태에서부터 동작시킬 수 있게 하는 모듈등으로 구성되어있다.

2) 프로그래밍 장치

유저와 PC의 시퀀서 사이의 정보교환을 원활하게 해주는 장치로, 유저-프로그램을 유저-프로그램 메모리(RAM)영역에 입력시키는 일 및 유저-프로그램을 디버깅하는 일과 원하는 데이터를 디스플레이 해주는 일을 수행한다.

3) 입력, 출력, 타이머, 카운터 모듈

입력 모듈에서는 외부에서의 입력신호를 시퀀서에 알맞는 신호로 변환시켜 주며, 출력 모듈에서는 시퀀서에서 처리한 결과를 기계나 공정에 알맞는 신호로 변환시켜 제어하며, 타이머 모듈은 어떤 사건의 발생 회수를 헤아릴 필요가 있을 때 사용된다.

이러한 PC는 입력/출력 모듈이 모듈화되어 있어, 제어대상에 따라 모듈을 선택할 수 있으며, 유저-프

로그래를 변경하여 공정을 변경할 수 있는 프로그램어 빌리티가 있다.

또한, 고장 발견이 용이하고, 신뢰도(reliability)가 크며, 사용자가 컴퓨터에 대한 지식이 부족하더라도 편리하게 사용할 수 있으며, 또 컴퓨터에 비해 현장 투입성이 크다. 그리고 PC는 conveyor system, machine tools, automated multiwelders, paint spraying, assembly lines와 같은 일반적인 공정의 제어 기로 이용할 수 있다.

III. 하드웨어의 설계^[8,9]

하드웨어의 모듈은 S-100 bus를 통하여, 배터리 백업된 CMOS RAM 및 정전 복구 기능을 갖는 모듈을 포함하는 시퀀서와 인터페이스되는데, 하드웨어의 구성요소는 프로그래밍 장치, 입력모듈, 출력모듈, 타이머 모듈, 카운터 모듈로, 이들에 대한 자세한 설명은 아래와 같다.

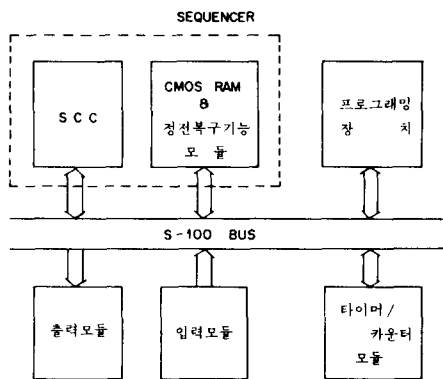


그림 2. 하드웨어의 구조 구성도
Fig. 2. Hardware configuration.

1. 프로그래밍 장치

프로그래밍 장치에는 사용자가 유저-프로그램을 구현하기 위한 인스트럭션을 구성하는 키(key)들 및 데이터 어드레스 키가 있고, 유저-프로그램을 시퀀서의 유저-프로그램 메모리 영역에 입력시킬 때 필요한 키와 프로그래밍을 할 때 편리하도록 하는 키가 있고, 사용자가 원하는 내용을 디스플레이하도록 하는 키와 디스플레이 장치가 있으며, PC의 동작을 관장하는 오퍼레이션 키가 있는데, 이 프로그래밍장치는 Intel 8279(programmable keyboard/display interface)를 중심으로 제작하였는데 구성도는 그림 3과 같으며, 동작원리는 아래와 같다.

Intel 8279에 연결된 키 보오드의 키를 누르면

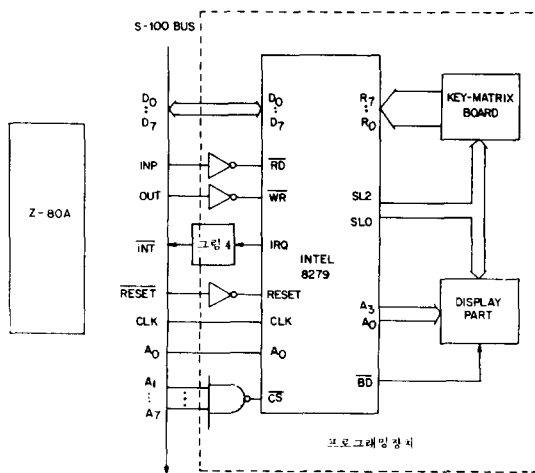


그림 3. 프로그래밍 장치와 SCC의 인터페이스
Fig. 3. Interface between programming device and SCC.

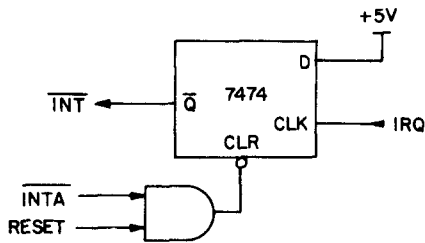


그림 4.
Fig. 4.

디바운스 타임 후 FIFO RAM에 키 포지션 정보를 받아들이고, IRQ 신호에 low-to-high 트랜지션을 발생시켜, 이 트랜지션이 D F/F를 트리거시켜 Q가 단자를 low 상태로 만들어 CPU로 인터럽트를 걸고, CPU가 인식하여 INTA를 내보내면 Q가 다시 high로 돌아간다. CPU와 8279사이의 정보교환을 위해서는 칩 선택트를 해야 하는데, 정보는 8279의 모드를 선택하기 위한 코맨드와 키포지션 정보를 8279로부터 받아들이는 데이터와 디스플레이를 위해 CPU가 8279로 보내는 데이터가 있다.

따라서, 코맨드 데이터의 구별은 어드레스 라인의 A₀를 사용하고 칩 선택트는 A₁~A₇을 사용한다. 8279로부터 키포지션 정보를 CPU가 받아 들일 때 누른 키의 포지션 정보를 순차적으로 받아들이도록 했으며, 디스플레이를 위해서는 8279의 출력중에서 A₀~A₃를 택하여, 원하는 데이터를 디스플레이 파트로 보

내고 스캔 라인을 사용하여 데이터를 디스플레이하는 세븐-세그먼트 LED를 선택하도록 했다.^[7]

2. 입력 모듈

입력신호로는 AC와 DC가 있는데, 그림 5에서 보는 바와 같이 모듈 선택회로에 의해서 버스에 연결된 입력 모듈 가운데 한 개를 선택하고, 상태 지시 LED에 의해 입력상태 점검과 고장진단을 쉽게 할 수 있으며, 포토-커플러에 의한 전원 분리로 노이즈 영향을 줄이고, 멀티플렉서를 사용하여 8개 입력중 한 개의 정보를 데이터 버스의 LSB(least significant bit)로 실는다.

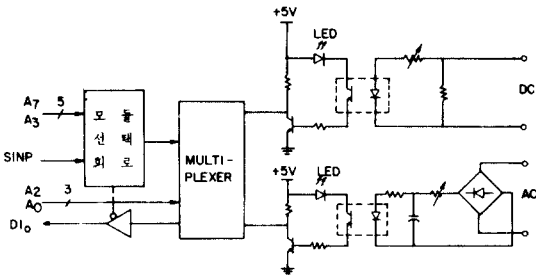


그림 5. 입력 모듈
Fig. 5. Input module.

3. 출력 모듈

그림 6에서 보는 바와같이 시퀀서에서 출력되는 논리 연산 결과를 DF/F에 래치(latch)시키며, DC 출력의 경우는 출력 트랜지스터를 세류레이션 또는 cut off시키고, AC 출력의 경우는 triac을 ON/OFF시켜 부하를 제어한다. 또 각각의 경우에 포토-커플러와 리

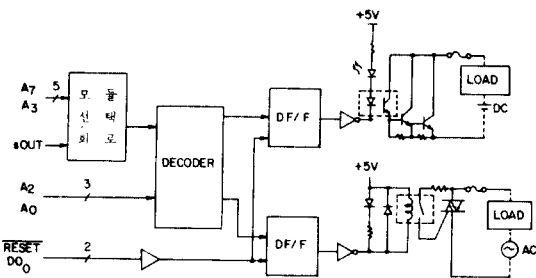


그림 6. 출력 모듈
Fig. 6. Output module.

드 릴레이(reed relay)를 사용하여 전원을 분리시켰으며, 상태지시 LED에 의해 출력상태 점검과 고장진단을 쉽게 할 수 있게 했다.

4. 타이머 모듈

시퀀스 제어도중 시간 간격 설정시 필요한 모듈로서, 회로구성은 그림 7과 같다. 시간간격에 해당하는 데이터를 다운 카운터에 입력시키고 A₈ 신호가 하이 레벨로 될 때 타이머가 동작을 시작하는데, 클럭의 선택에 의해서 0.1초~12.5초 또는 10초~1,250초 범위로 사용하도록 했으며, 10진 다운 카운터의 첨가로 랜덤한 클럭에 관계되는 타이머의 시간 정확도를 10배 증가시켰다.

다운 카운터는 클럭에 따라 동작하다가 출력 데이터가 00으로 된 후 OFFH로 되면서 DF/F을 세트시킨다. 이 DF/F은 설정시간 경과 여부를 나타내는데, DF/F의 출력을 데이터 버스의 LSB로 연결하여 입력 모듈과 컴퓨터블하게 했으며, 상태지시 LED로 쉽게 시간 경과 여부를 알 수 있다. 정전되었다가 전원이 복구될 때, 시스템을 정전전의 상태로 만들기 위해서는 정전이 될 때의 타이머 데이터를 시퀀서가 입력시켜 배터리 백업된 CMOS RAM에 저장하게 되는데, 이 데이터 중에서 다운 카운터의 출력 데이터가 데이터 버스의 D7~D1을 차지한다. 설정기간이 지난 후 타이머 데이터는 OFFH가 되고, 타이머를 리셋시키면 OFFH가 된다.

한 개의 타이머는 유저-프로그램에서 한번만 사용되어 왔는데, 유저-프로그램 상에 약간의 프로그램 첨가에 의해서, 시간 간격 설정이 각각 다른 시간에 일어날 경우, 한 개의 타이머를 여러번 사용하는 새로운 기능을 개발했다. 이때는 그림 12의 예와 같이 타

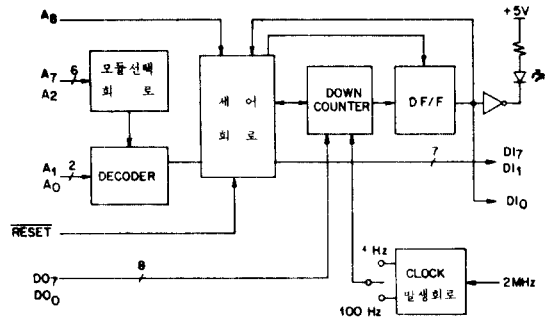


그림 7. 타이머 모듈
Fig. 7. Timer module.

이머를 리세트시키고, 시간 경과 여부를 나타내는 정보를 래치의 리세트 기능을 가진 내부 릴레이(internal relay)에 저장하고, 또 이 내부 릴레이를 한 스캔 타임 동안 지연시키는 유저-프로그램이 필요하다.

5. 카운터 모듈

클럭 입력을 제외하면 타이머 모듈과 거의 같다. 클럭은 유저-프로그램에 의한 내부 클럭과 외부에서 직접 연결 가능한 외부 클럭중 한 개를 사용하며, 계수 범위는 1-125로 했다.

6. CMOS RAM과 정전 복구기능 모듈

정전이 되더라도 시스템을 정전전의 상태에서부터 계속적으로 동작하도록 하며, 구성은 그림 8과 같다. 정전이 될 경우 전원검출회로에서 정전을 검출하여 non maskable interrupt(NMI)를 걸어 NMI 서어비스 루틴에서 정전으로 인해 없어질 타이머, 카운터의 현재 데이터를 배터리 백업된 CMOS RAM에 저장한 후 이 RAM을 디스에이블시키기 위한 신호를 내보내면, 서트다운 회로에서 RAM을 디스에이블시키게 되어 저장된 데이터가 변화없이 저장된 상태를 유지하게 된다. 전원이 복구될 때는 전원스위치 위치 검출회로로부터의 전원 스위치 위치 정보를 받아들여 PC 시스템을 정전 전의 상태로 회복시키거나, 처음부터 다시 동작시키는 프로그램을 수행시킨다.

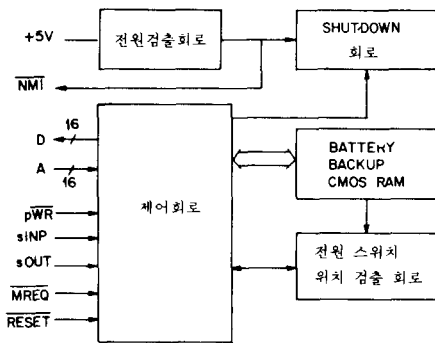


그림 8. COMS RAM과 정전 복구기능 모듈
Fig. 8. CMOS RAM and power failure recovery module.

IV. 소프트웨어의 설계^[8,9]

소프트웨어의 전체적인 구조는 PC를 인니시얼리제이션 한 후, 모니터 프로그램에서 유저- 프로그램을 유저-프로그램 영역에 입력하는 일과, 유저가 원하는 내용을 디스플레이 하도록 하는 일 및 새로운 유저-프로그램 입력을 용이하게 하는 일을 해주며, 유저가 원

하는 때에 execution program을 수행하도록 해준다. 또한 부수적으로 갑작스러운 정전에 대해 시스템을 정전전의 상태로 복구시키는 데 필요한 정전복구 프로그램이 있다.

이에 대한 플로우차트는 그림 9와 같으며, 각각의 자세한 내용은 아래와 같다.

1. 인니시얼리제이션 프로그램

SCC 자체내의 인터럽트 소오스가 있기 때문에 불필요한 인터럽트를 방지하기 위해 마스킹을 하며, 프로그램 디바이스의 8279를 인니시얼리제이션 한다. 즉 8279의 여러 모드 중 필요한 모드를 선택한다. 또한 유저-프로그램 포인터 인니시얼리제이션, 인터럽트 모드를 선택한다.

2. 모니터 프로그램

모니터 프로그램은 키이 포지션 정보를 받아들여 분류한 후, 해당하는 오퍼레이션을 수행하는 프로그램으로 그림10과 같으며, 수행하는 일에 따라서 서브루틴으로 나누어 구성했는데, 그 내용은 아래와 같다.

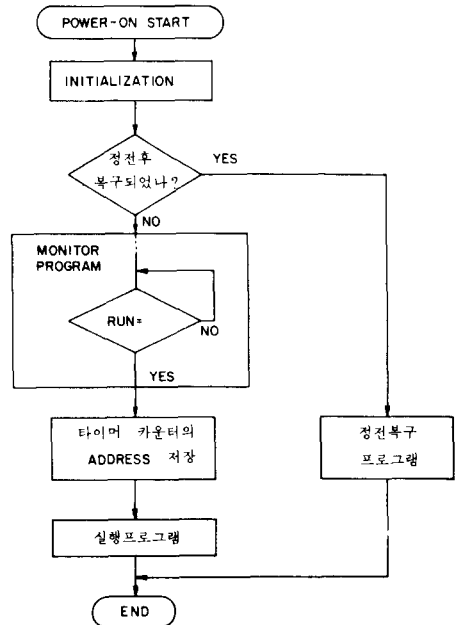


그림 9. 전체적인 소프트웨어의 구조
Fig. 9. Flowchart of overall software.

1) 데이터 / 어드레스 키이 서브루우틴

유저-프로그램 영역에 저장하는 데이터 또는 어드레스는 1byte인데, high nibble인지 또는 lower nibble인지를 판별해야 하며, 리턴 키이를 치면 데이

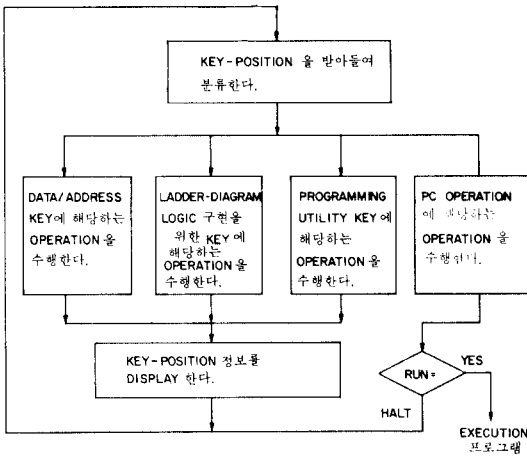


그림10. 모니터 프로그램의 플로우차트
Fig. 10. Flowchart of monitor program.

타 또는 어드레스가 입력된다.

2) 레더 다이어그램 로직 구현 키 서브루틴

레더 다이어그램 로직을 구현하기 위해서 RD, AND, OR, RDSTK key들과 CR, NOT key들의 조합과 ANDSTK, ORSTK, TMR, CTR, OUT, OUTNOT, CRO key와 유저-프로그램의 끝을 표시하는 END 키를 사용하는데, 각각의 인스트럭션을 나타내는 키들의 포지션 정보를 받아들여 인코딩을 한 후, 리턴 키를 치면, 유저-프로그램 영역에 입력한다.

3) 프로그래밍 유틸리티 키 서브루틴

더리-트(delete)와 디스플레이와 리세트가 있는데, 더리-트는 유저-프로그램의 1 어드레스를 클리어하고, 리세트는 유저-프로그램 영역을 전부 클리어 한다. 디스플레이는 유저-프로그램의 어드레스와 데이터를 디스플레이 한다.

4) PC 오퍼레이션 키 서브루틴

RUN과 HALT가 있는데, RUN은 execution 프로그램을 수행하도록 하고, HALT는 고장 발생이나 공정 변경시 필요한 데, execution 프로그램의 수행을 중단시킨다.

3. 실행 프로그램과 스킵기능

실행 프로그램은 제어 대상으로부터 들어오는 신호를 로지컬 프로세싱하여 그 결과를 출력시키는 프로그램으로, 실행 프로그램을 고려함에 있어서 스캔 타임을 줄이는 방법의 하나로 스킵 기능을 고려했는데, 이는 PC의 특성과 시퀀스 제어의 특성을 조합 이용한 것이다.

1) 실행 프로그램

Execution 시 로지컬 프로세싱은 유저-프로그램을 구성하고 있는 1 block(인스트럭션과 어드레스 또는 인스트럭션과 어드레스 및 데이터) 단위로 fetch하여 디코딩한 후, 디코디드 인스트럭션에 해당하는 서브루틴을 수행하므로써 이루어진다. 이때 유저-프로그램을 캐치(fetch)해오는 방법은 인덱스드 어드레싱 모드를 사용하고, 디코딩하는 방법은 테이블 방식을 사용한다. 또한 프로그래밍 장치로부터 HALT 신호가 들어오지 않는 한 유저-프로그램의 마지막인 END가 나오면, 다시 유저-프로그램의 처음부터 캐치해오는 cyclic operation을 하도록 했는데, 플로우 차아트는 그림11과 같다.

2) 스킵 기능

스킵 기능은 absolute 노드(그 점에서의 어큐블레이터의 내용이 뒤에 해당하는 로지컬 프로세싱에 관계없이 한 rung의 출력을 지배하는 노드)에서는 OUT이나 OUTNOT나 CRO나 타이머나 카운터 인스트럭션이 유저-프로그램에서 나올 때까지는 서브 루틴을 수행하지 않고, 유저-프로그램만을 스캔한다. 그 방법에 있어서는 RDSTK이나 RDSTKNOT이나 REST KCR이나 RDSTKNOTCR 서브루틴을 수행할 때에는 플래그 레지스터(스킵 결정을 위한 플래그 레지스터)를 1개씩 증가시키고, ANDSTK이나 ORSTK 서브루틴을 수행할 때에는 플래그 레지스터를 1씩 감소시킨다. 따라서 AND나 ANDNOT이나 ANDCR이나 ANDNOTCR 서브루틴을 수행한 결과가 0이고, 플래그 레지스터의 내용이 0이면 스킵하고, RDSTK이나 RDSTKNOT이나 RDSTKCR이나 RDST NOTCR의 서브루틴에서 플래그 레지스터의 내용을 1씩 증가시킨 후, 플래그 레지스터의 내용이 1이고, 해당 서브루틴 전단까지의 로지컬 프로세싱 결과가 0이면 스킵한다.

4. 정전 복구 프로그램

프로그래밍 장치에서의 런 키가 눌러지면, NMI 서어비스 루틴에서 사용할 수 있도록 유저-프로그램에서 사용된 타이머와 카운터의 포트 어드레스와 사용된 갯수를 저장시키고, 데이터나 카운터의 출력 데이터를 OFFH로 만들어 리세트 상태에 있게 한다.

NMI 서어비스 루틴에서는 CPU 레지스터들을 스택에 저장한 후, 앞에서 저장된 타이머, 카운터의 갯수와 포트 어드레스를 이용하여 타이머, 카운터의 출력 데이터를 배터리 백업된 CMOS RAM에 저장하고, 이 RAM을 디스에이블시키게 하는 신호를 보낸다.

PC 시스템에 전원이 공급되면 인니시얼라이제이션 프

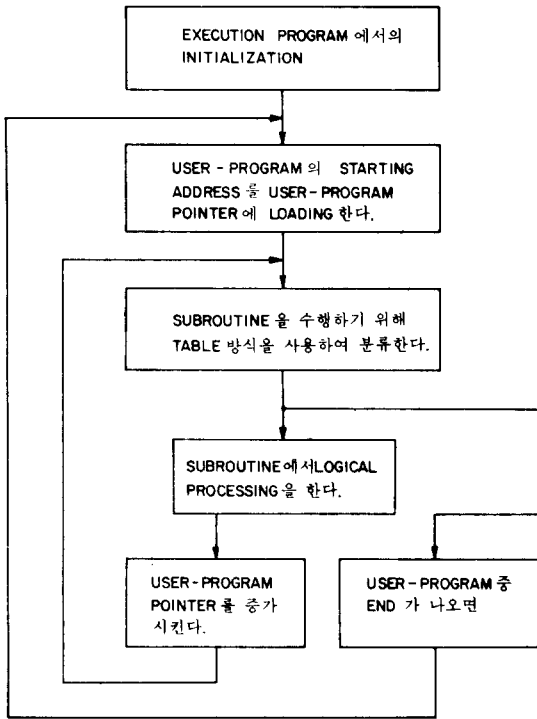


그림 11. 실행 프로그램의 플로우차트
Fig. 11. Flowchart of execution program.

로그래밍 실행후 전원 스위치 위치 신호에 따라 모니터 프로그램 또는 정전 복구 프로그램을 실행하게 되는데, 정전 복구 프로그램에서는 타이머, 카운터, CPU 레지스터들을 전부 정전 전의 상태로 복구시킨다.

V. 실험결과 및 검토

제어대상을 드릴링 머신인 N/C 100Trainer set와 소규모 콘베어 시스템으로 잡아 시퀀스 제어 실험을 행했다. 드릴링 머신은 ±X축 모터, ±Y축 모터(공작대 위치 조정), ±Z축 모터(드릴 위치조정)와 드릴을 돌리는 스피들 모터를 가지고 있으며, 스타트 버튼을 누르면 콘베어가 움직이기 시작하고, 운반되어 온 공작물을 공작대에 붙린 후 드릴링 머신의 스타트 버튼을 누르면 지정된 위치에 구멍을 뚫게 되는데, 리미트스위치와 한 개의 타이머를 여러 번 사용할 수 있는 기능을 사용하여 모터에의 전원 공급시간 조정으로 위치를 잡도록 해서 실험하여 만족스런 결과를 얻었다. 간단한 실험의 예로서 한 개의 타이머를 사용하여 램프를 번갈아 1초간 on, 2초간 off시키는 시퀀스 제어에 대한 유저-프로그램이 그림12에 도시하였다. 한 개의 타이머를 여러번 사용하게 될 때 유저 프로그램이

길어져 복잡해지고, 메모리와 내부 릴레이를 많이 차지하게 되지만 타이머를 줄일 수 있다는 잇점이 있으며, 특히 한 두 개의 타이머에 의해 타이머 모듈이 필요할 경우 유용하게 사용될 수 있으리라 생각된다. 또 유저-프로그램의 각 단이 간단히 이루어 질 때는 스킵 기능의 효과가 없어지는데, 보다 효율적인 스킵 기능의 개발이 바람직하다.

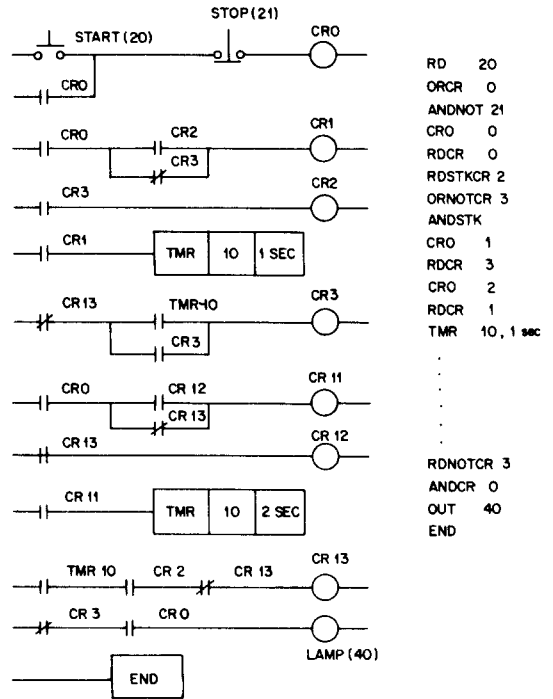


그림 12. 간단한 유저-프로그램 예
Fig. 12. A simple example of user program.

VI. 결 론

본 논문 연구에서 설계 제작한 시퀀스 제어용의 PC는 입력/출력 및 타이머, 카운터의 접점수가 240개이고, 내부 릴레이는 256개이고, 유저-프로그램 영역은 2 Kbytes이며, 스캔 타임의 최대치는 40 msec/2 Kbytes이고, 정전후 전원이 복구될 때 정전 전의 상태에서부터 연속적으로 제어할 수 있다. 특히 한 개의 타이머를 여러번 유저-프로그램에서 사용할 수 있는 기능과 스캔 타임을 줄이는 방법의 하나로 스킵 기능을 연구했다.

본 연구에서 제작한 PC가 현장에 쓰여지기 위해서는 신뢰도를 고려하여 high threshold gate를 써야겠고, 스캔 타임을 더욱 줄이기 위해 하이 스피드 프

로세서를 쓰거나 비트-슬라이드 마이크로프로세서 아키텍처의 적용을 연구해야 하며, 그 기능을 더욱 다양화하여 아날로그 신호를 다룰 수 있도록 하고, 아울러 PID 콘트롤러 기능을 추가하는 것이 바람직하다고 사료된다.

參 考 文 獻

- [1] S.J. Bailey, "The programmable controller is finding its in-line niche," *Control Engineering*, pp. 51-55, Feb. 1980.
- [2] E.J. Kompass, "New programmable controller has dual microprocessors", *Control Engineering*, pp. 55, Oct. 1978.
- [3] H.M. Morris, "Implementing PID in PCs", *Control Engineering*, pp.88-90, Oct. 1981.
- [4] V.J., Maggioli, "Programmable controllers in process control applications," *IEEE Trans. Industry Applications*, vol. IA-15, no. 6, Nov./Dec. 1979.
- [5] G. Faust, "Programmable controller offers fiber optic data link for remote I/O", *Control Engineering*, pp. 53-54, Oct. 1979.
- [6] D.M. Cherbam "Programming PCs to detect faults in machines or processes," *Control Engineering*, pp. 57-61, Feb. 1981.
- [7] "Peripheral Design Handbook," Intel.
- [8] 김용수, *Software and Programming Device for Programmable Controller.*, KAIST 석사논문, 1983.
- [9] 김영현, *A Study on Hardware Modules for a Programmable Controller.*, KAIST 석사논문, 1983.
- [10] *PC Course. Instruments and Control Systems* 1981. 2 - 1981. 10.