

현장 기술자를 위한

소형 PC에 의한 시퀀스 제어

현·장
기·술

2

역 / 박 한 중 (협회 교육홍보위원)

제2장 PC를 동작시켜 본다

1. 여하간에 움직여 본다

프로그래머블 컨트롤러(PC)에 대한 개설은 제1장에서 언급한 바 있지만 그 밖에도 알아 두어야 할 것이 많이 있다. 그러나 처음에 많은 지식이 주어 지더라도 실제로 이 지식을 사용하여 행동하는 기회가 없으면 지식은 살려지지 않는다.

그러므로 지식을 설명받기 전에 실제로 PC를 사용하여 무조건 자신의 손으로 움직여 보도록 하자. 동작시켜 보고 PC가 어떠한 움직임을 하는가를 확인하면서 PC의 사용방법을 습득하도록 하는 것이 좋다고 생각한다.

명령을 사용하여 프로그램을 프로그램 콘솔에서 키 인하고 기록한 프로그램을 확인하거나 수정하거나 하여 실행할 때까지의 일련의 조작방법에 대해서 기술하기로 한다.

이 조작방법은 PC의 기종에 따라 다소 달라지므로 일반적으로 기술할 수는 없다. 그러므로 어느 하나의 기종을 선정하여 이것에 대한 조작방법을 구체적으로 기술한다. 하나의 조작방법을 마스터하면 다른 기종에 대해서도 그 PC에 달려 있는 취급설명서나 테크니컬 가이드 등을 보면서 독자적으로 조작방법을 습득할 수 있게 되므로 이와 같은 방법을 택하기로 하였다.

여기서 선정한 기종은 초소형이며 가격도 적당

하고 교육·훈련용으로 제작된 것으로서, PC 본체, 입출력 유닛, 전원 및 입력조작용 스위치 BOX가 1세트로 된 것이다. 그러나 이대로도 산업용으로 실제 제어에 사용할 수 있는 것이다.

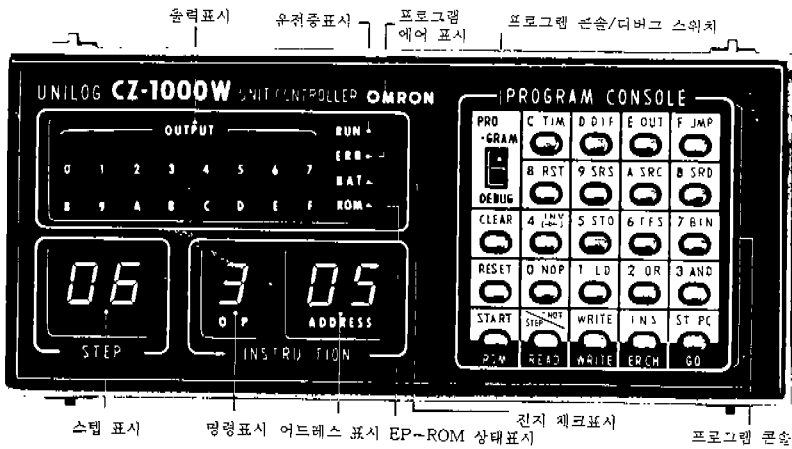
2. 키 인 방법

전원 플러그를 콘센트에 꽂고 전원을 투입하기 전에 프로그램 콘솔의 각 부분에 대해서 설명하기로 한다.

그림 2.1 우측에 0~F까지의 16개와 기타 몇 개의 푸시버튼 스위치(키)가 배열되어 있다. 이 부분을 프로그램 콘솔이라고 하며, 키를 누름으로써 PC에 프로그램을 입력하거나 기타 여러 가지 명령을 부여할 수 있게 되어 있다.

그 좌측 아래 쪽에 숫자 등을 표시하며 적색으로 점등하는 곳이 있다. 이 부분을 명령표시부라고 한다. 명령표시부는 STEP과 INSTRUCTION의 두 가지로 구성되고 INSTRUCTION은 OP와 ADDRESS로 구성되어 있다.

STEP이란 프로그램의 스텝 표시를 하는 부분으로서 지금 06이라고 표시되어 있는데. 이것은 06 스텝 상태에 있고 그 때의 INSTRUCTION, 즉 명령이 3 05 * 인 것을 표시하고 있다. 3이 표시되어



〈그림 2.1〉 프로그램 콘솔의 외관

있는 부분을 OP(Operation), 05가 표시되어 있는 부분을 ADDRESS라고 부른다. OP는 명령(후술하는 AND, OR 등)을 16진수의 숫자로 부호화하여 표시하고 있다. ADDRESS는 번지(어드레스)로서 명령으로 표시된 처리 장소나 수치 데이터를 표시하는 부분이다.

이 윗 쪽에 OUTPUT라고 표시된 곳이 출력표시부이다. 이 경우 0~F까지 16점의 출력상태가 표시된다. 발광 다이오드(LED)가 점등되어 있는 개소가 출력상태가 「H」 레벨**(High Level)로 되어 있는 것을 나타내고 있다.

출력표시 우측에는 RUN: 운전중 표시, ERR: 프로그램 에러 표시, BAT: 전지 체크 표시, ROM: EP-ROM상태 표시의 각각을 표시하는 LED가 있는데, 이것들은 PC를 운전해 보고 그 때마다 설명하기로 한다.

3. 콘솔의 조작

PC에 전원을 넣고 PROGRAM CONSOLE부의 PROGRAM/DEBUG(프로그램 콘솔/디버그 스위치)를 PROGRAM측으로 한다. 그러면 STEP(스

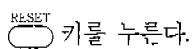
텝)과 INSTRUCTION(명령) 표시부의 LED가 다음과 같이 된다.



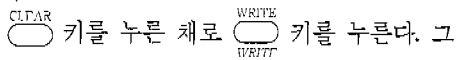
INSTRUCTION부의 ※ ※※ 는 0부터 F까지의 어떤 숫자, 문자가 표시되는데 그것은 정해진 것은 아니다. ※※ 등으로 되어 있는 곳은 어떤 숫자, 문자가 표시되면 그것이 무엇인가는 특별히 생각할 필요는 없다(프로그램의 내용을 강제적으로 지운 경우에는 0 00 이 된다).

그러나 지금부터 새롭게 프로그램을 기록할 때는 앞의 프로그램이 있으면 불편이 생긴다. 그래서 RAM에 있는 불필요한 내용의 프로그램을 지울 필요가 있게 된다. 이 때는 다음과 같이 한다.

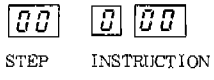
우선



다음에




그러면 STEP과 INSTRUCTION의 LED 표시가 바뀌고 수 초에 다음과 같이 된다.







* OP부의 3을 니모닉 코드로 그대로 AND 등으로 표시하는 기능이 있는데, 이것은 단순히 16진 코드(숫자)를 문자로 나타낸 것뿐이다.
 ** 2차신호의 상태를 전압의 고저로 각각 고(高)는 「H」, 저(低)를 「L」로 표시하고 있다. 「H」 레벨이란 전압의 고위상태에 있고 주력 유닛의 릴레이가 동작상태로 되어 있는 것이다.

이상은 프로그램을 기록하기 위한 준비이고, 여
기부터가 드디어 프로그램의 키 인이다.

















 키를 누른다.

그러면 INSTRUCTION부의 모든 LED가 지워
진다. 계속해서 다음 순서로 키를 누른다.

이하 그 때의 LED 표시를 나타낸다.












[누르는 키] [누른 다음의 LED 표시]





			
			
			
			
WRITE	STEP	INSTRUCTION	



지금 하고 있는 것을 설명해 둔다. STEP부의 2
자리의 표시는 프로그램 스텝 번호를 나타낸다.
INSTRUCTION부의 3자리 표시는 프로그램 스텝
번호에 기입되어 있는 내용이다. 프로그램 스텝
번호 00번지에 1, 즉 LD 명령(Load의 생략형, 후
술)과 그 데이터 00을 기록한 것이 된다.






단, 이 00 스텝 번호의 00, INSTRUCTION부의
OP부 표시의 1, 등 ADDRESS부의 00은 보통 우
리들이 사용하고 있는 10진수의 00이나 1이 아니
고 16진수의 표시로 하고 있다. 그러므로 0 ~ 9
의 숫자 이외에 A B C D E F 등의 문자 표시도
사용되므로 하여간에 여기서는 숫자의 표시방법이
약간 다르다는 것을 머리에 넣어 두기 바란다.





















여기서, 경우에 따라서는 키를 잘못 누르는 일
도 있다고 생각되므로 이것에 대해서 언급해 두기
로 한다.

예를 들면 02 스텝으로    라고
눌러야 할 것을    로 눌러 버렸다면
 키를 누르기 전이면 다시 한번  키를
누른다. 그러면 INSTRUCTION부의 LED가 전부
지워지므로 계속해서 바르게    로
누르면 된다.


만일 불행히도  키를 눌러 버렸다면  키
를 누른다. 그러면 STEP, INSTRUCTION부 모두
LED가 지워지므로 계속해서   와 같이
스텝 번호를 누르면 02 스텝의 표시와 그 스텝의
내용이 LED에 표시된다.

그러면 앞의  키를 누르지 않았을 때의 조
작방법과 동일하게  키를 누르고 바르게 키 인
한다. 이것으로 잘못 키를 조작했을 경우의 설명
은 끝이다.


본론으로 돌아가서 키 조작의 설명을 계속한다.
     까지 키 인되어 있
다. 동일하게 아래의 키 조작을 하여 본다.

				
FR CH				WRITE
				
ER CH				WRITE
				
LR CH				WRITE
				
LR CH				WRITE

이상으로 키 입력은 끝이다. 그러나 자신도 모
르는 미스를 하고 있는지도 모른다. 그러므로 여
기서 확인을 하여 두자.

 키를 누른다. 그러면 다음과 같이 00 스텝
과 그 내용이 LED에 표시된다.

		
STEP	INSTRUCTION	

다음에  키를 누른다. 그러면 다음의 01 스
텝과 그 내용이 LED에 표시된다. 이하, STEP 키
를 1회 누를 때마다 스텝이 하나씩 증가하고 최종
의 04 스텝까지 계속한다. 그리고 키 인하여야 할
내용과 LED의 표시가 일치하고 있는가의 여부를
확인한다.

미스가 없으면 다음으로 진행하자. 만일 불행히
도 미스가 발견되면 전술한 방법으로 바르게 고쳐
둔다.

이것으로 키 인한 것의 확인이 종료되었다. 지
금까지 행한 것은 PC에 프로그램을 기록하는 것

이었는데 그 작업이 완료된 것이다. 그러면 다음으로 드디어 이 프로그램을 실행해 보자.

4. 디버그와 운전

프로그램 실행 전에 조작용 스위치 박스(OPERATIONAL BOX)의 커넥터를 OP. CONECTOR부에 접속한다. 이 조작용 스위치 박스에는 스텝 스위치가 16개(00~0F)까지 배열되어 있다. 그 스위치를 전부 OFF측(ON이라고 표시되어 있지 않은 쪽)으로 한다.

조작용 스위치 박스의 스위치군에는 INPUT라고 표시되어 있는 것과 같이 테스트를 위한 입력용 스위치로서 사용하는 것으로, 실제로 제어하는 기기를 운전하기 전에 PC 출력측 단자에 기기를 접속하지 않고 시운전하기 위해 사용되는 것이다. 당연한 일이지만 조작용 스위치 박스의 접속선은 입력 유닛으로 되어 있다.

프로그램 콘솔/디버그 스위치를 DEBUG측으로 한다. 그러면 STEP, INSTRUCTION부의 LED가 다음과 같이 된다.



START ROM 버튼을 누르면 운전중 표시(RUN)의 LED가 점등되고 STEP INSTRUCTION부의 LED가 다음과 같이 바뀐다.



여기서 조작용 스위치(INPUT)의 01번을 ON측으로 한다. 입력 유닛의 1회로의 LED(01번지에 대응)가 점등하고 있지만 기타는 아무런 변화가 없을 것이다. 그러나 이 조작은 의미가 있는 것이다(뒤에 알게 된다).

다음에 INPUT 00번지의 스위치를 ON으로 하고 곧 다시 OFF로 한다. 어떻게 되는가? “짹짹” 하는 소리가 나고 입력 유닛의 1회로의 LED(00번지에 대응)가 일단 점등되고 곧 다시 꺼지며 INS-

〈표 4.1〉 자기유지회로 프로그램

스텝 No.	명 령		비 고
	OP *	ADD *	
00	LD	00	
01	OR	10	
02	AND	01	
03	STO	10	
04	OUT	E0	

(주) * OP: Operation(조작, 니모닉코드로 표기)
* ADD: Address(번지)

TRUCTION의 OP부 LED도 「L」에서 일단 「H」가 되고 곧 다시 「L」이 된다. 그러나 콘솔의 OUTPUT부 LED와 출력 유닛군 좌단의 Xi회로(E0번지에 대응)의 LED가 점등한 채로 있을 것이다.

그러면 INPUT 00번지의 스위치를 OFF로 하고 곧 다시 ON으로 한다. 그러면 “짹짹” 하는 소리가 나고 출력 유닛의 LED가 꺼질 것이다. 이것을 확인하도록 하자.

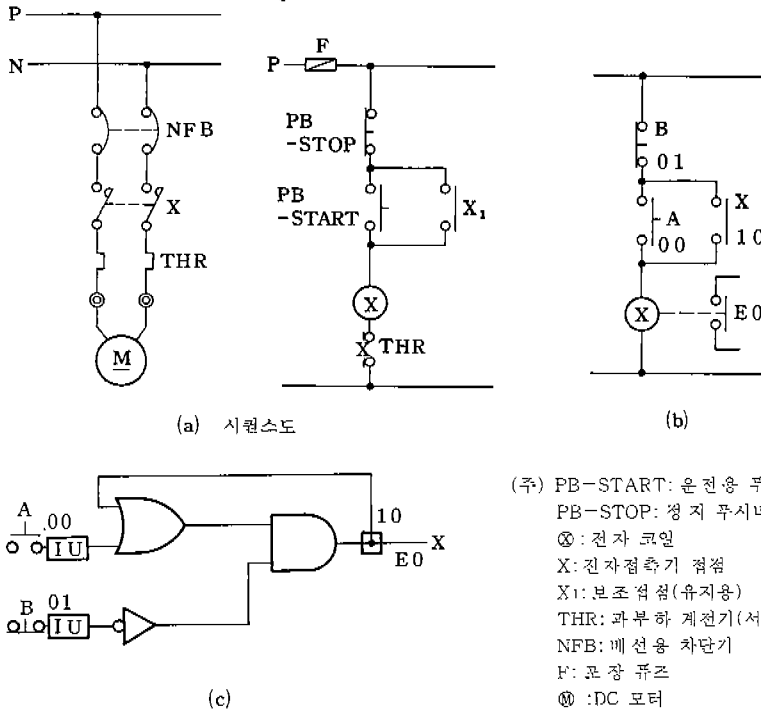
만일 상기와 같이 되지 않을 때는 프로그램의 키 인이 어딘가 틀려 있으므로 다시 한번 보고 재차 실행해 보도록 한다.

여기까지의 동작을 보면 이미 알았을 것으로 생각되지만 이 동작은 자기유지회로 그것이다.

그림 4.1에 릴레이 시퀀스도로 자기유지회로를 들었다. (a)는 모두가 알고 있는 회로로서 특별히 설명할 필요가 없는 것이다. (b), (c)는 PC로 제어하기 위한 프로그램용으로 번지를 할당한 것이다. 표 4.1이 이 자기유지회로의 프로그램 리스트이다.

프로그램의 내용에 대한 상세한 것은 제4장 이하에서 기술하므로 여기서는 간단히 PC의 동작과 입출력 관계를 설명해 둔다. 다시 한번 프로그램을 실행하여 조작 스위치를 조작한 것을 상기해 보자. 조작 스위치의 01 점점에 관해서 이것은 실제 운전시에는 수동조작 자동복귀의 b점점이기 때문에 상시 페로가 되도록 INPUT 스위치를 ON으로 하여 두었다.

한편, 조작 스위치의 00 점점은 실제로는 수동조작 자동복귀의 a점점이기 때문에 상시 게로가 되



<그림 4.1> 자기유지회로

고 INPUT 스위치는 OFF로 하여 두었다.

운전할 때는 00이나 01 점접의 실제 동작의 모의 때문에 조작용 스위치는 일단 ON으로 하여도 곧 다시 OFF하는 조작을 하였다.

그러면 다시 한번 조작용 스위치를 조작하여 자기유지회로의 동작을 확인하기 바란다. 동작내용이 이해되었을 것으로 생각한다. 이 조작은 디버그* 라고 하며, 프로그램에 틀림이 없는가, 키 인이 바르게 되어 있는가 등을 체크하고 만일 프로그램의 논리 오차나 미스 키 인이 있을 때는 이 단계에서 올바른 프로그램으로 만들어 둔다.

디버그가 완료됐으므로 실제 기계를 시운전하기로 한다. 여기서는 본 트레이닝세트를 사용하여 설명하고 있는 관계로 이 PC 시스템으로 구성할 수 있는 입출력 기기를 선택한다. 입력 유닛은 유접점 신호입력을 받는 전압 변환용의 것이다. 또한 출력 유닛은 부하의 종류가 솔레노이드, 밸브 등의 유접점 출력 개폐용이고 부하는 AC 100V-1A,

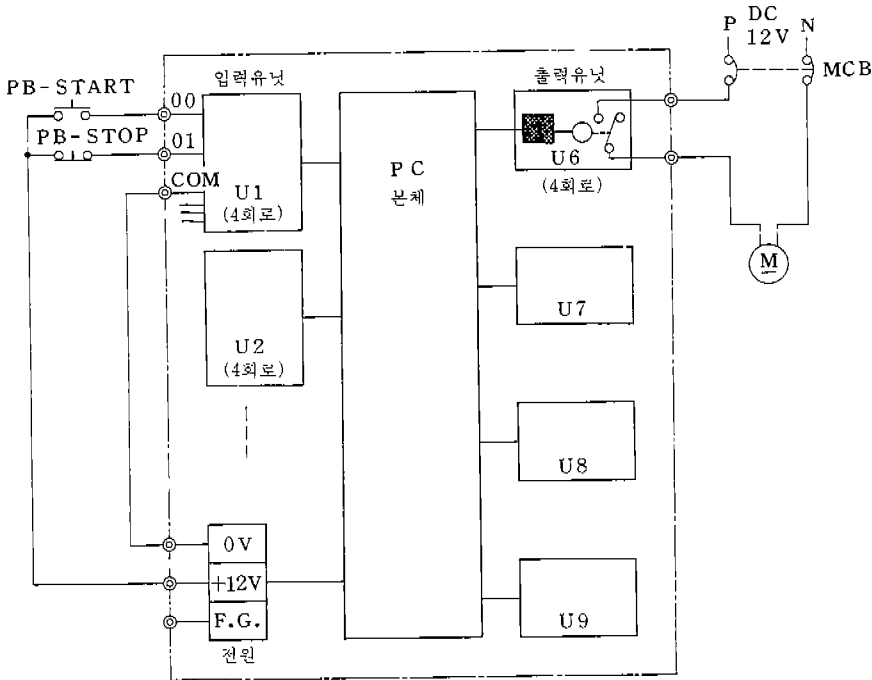
DC 24V-1A까지 접속할 수 있는 것이다.

모터를 움직이는 경우의 출력 유닛의 바른 선정은 모터 구동용 출력 유닛군이 되는데, 본 트레이닝 세트에 이미 장비되어 있는 소켓은 모터 구동용 출력 유닛에는 적합하지 않다. 이 때문에 편의상 유접점 출력용 출력 유닛을 사용하기로 하는데 안전을 고려하여 모터는 DC 12V, 10 수W 정도의 것을 무부하로 운전하기로 한다.

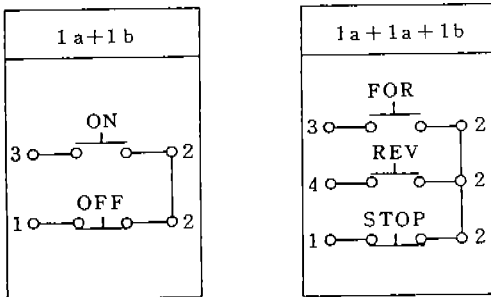
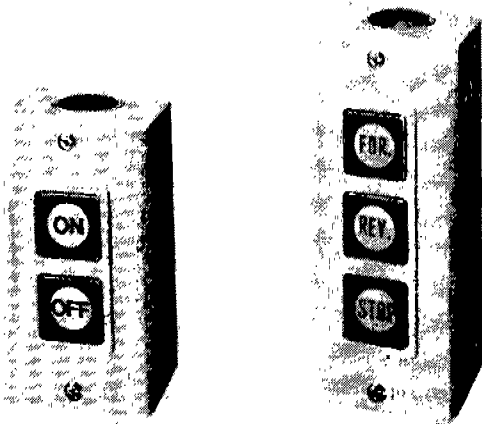
그림 4.2와 같이 PC의 입출력 단자와 푸시버튼 스위치, 모터, 전원을 접속한다. 여기서 주의할 것은 PB에는 수동조작 자동복귀 점접을 사용하는 것과 PB-STOP은 b점접인 것이다. 예를 들면 그림 4.3과 같은 점접 구성의 것을 사용하는 것이 좋을 것이다.

부하의 모터로의 회로 구성은 그림 4.1(a)에서는 P, N 양 극을 개폐하도록 전자점촉기 점접을 접속하고 다시 또 서멀 릴레이(THR)도 달려 있었지만 이것을 그림 4.2의 접속도로 변경하였다. 이유

* 해충은 제거한다는 의미에서 컴퓨터 용어에서는 프로그램의 오차(버그)를 발견하여 수정하는 것.



〈그림 4.2〉
PC로의 입출력기기의 접속에



〈그림 4.3〉 푸시버튼 스위치

는 몇가지가 있지만 이것은 뒤에서 설명하기로 한다.

그러면 이제 모터를 운전하기로 하자. 모터 구동 전원측의 MCB를 ON으로 하고 PC의 프로그램을 움직여 본다. PB-START의 푸시버튼 스위치를 누른다. 모터가 운전을 개시한다. 다음에 PB-STOP의 푸시버튼 스위치를 누르면 모터는 정지한다.

이상 이 장에서는 제어내용은 어찌되었던 간에 PC를 움직여 보는 것에 주안점을 두고 설명하였다. PC는 잘 움직였는지? 프로그램으로 모터가 제어된 것은 아마도 전자 릴레이 제어에는 없는 그 무엇을 느꼈을 것으로 생각한다.

다만 여기서 주의해야 할 것은 PC가 움직였다고 하지만 제어 원리나 프로그램 내용을 충분히 이해하고 움직인 것은 아니라는 것이다. PC의 조작방법과 입출력의 구성이나 접속방법에 대해서만 알아 본 것이다.

다음 장부터는 제어 원리라고 할 수 있는 논리 회로와 프로그램 내용에 대한 설명을 순서적으로 하기로 한다.

다음호에 계속