

記憶再生式 産業用로봇의 制御部 設計에 관한 研究

論 文
29-7-2

A Study on the Design of Control Unit for Playback-type Industrial Robot

宋 祥 燮* · 金 昇 弼** · 卞 增 男***
(Sang-Sup Song, Seung-Pil Kim, Zeungnam Bien)

Abstract

The design of a control unit for a playback-type industrial robot is studied. Implemented for the cylindrical-coordinate type industrial robot with 5 degrees of freedom, the control unit constructed for the study consists of (i) z-80 μ p-based μ -computer control system (ii) Teach-Box for work command, and (iii) various softwares for generating signals for servo driving unit and operating the robot as playback-type. Softwares are developed by using high level Basic Language and low level z-80 Assembly Language for ease of programming and speed of program execution. To show the effectiveness, an example is included.

1. 서 론

1960년대 초 미국의 G.C. Devol의 제안에 따라 Unimate 회사에서 GM 자동차 차체조립과정에 점용접 (spot-welding)용으로 처음 사용되기 시작한 산업용 로봇트는 단순반복작업, 약조건하의 작업, 위험한 작업등의 여러 분야에서 효율성을 인정받고 있다. 더욱이 고도로 발달된 LSI 기술에 의한 마이크로 프로세서의 도입하여, 저렴한 가격으로, 여러 종류의 작업을 프로그램의 간단한 수정에 의해서 행할 수 있는 다기능성(versatility)을 갖출 수 있게 되어, 소위 유연한 자동화(flexible automation)의 구현에 박차를 가하고 있다. 일본, 미국, 서구등의 선진국가에서는 급속도로 로봇트에 대한 연구개발이 진행되고 있으며 현재

30,000여대의 각종 로봇트가 자동차산업, 공작기계산업, 조선업등의 여러 분야에서 널리 사용되고 있다. [19, 20, 21] 이러한 추세에 따라, 최근 국내에서도 산업용로봇트에 관한 관심이 고조되고 있고 여러기관에서 계속해서 연구사례를 발표하고 있다. 국내의 몇몇 기업체, 즉, 기아산업, 대우 중공업등에서 시험용으로 산업용로봇트를 도입 사용중이며 이의 국산화에 대한 관심이 높아가고 있다.

본 논문은 한국과학재단의 재정적지원으로 한국과학원 제어연구실에서 연구개발한 교시형로봇트를 설계 제작한 결과를 두 편으로 나누어 수록한 것으로, 제작한 로봇트는 다이캐스팅, 열처리과정, 또는 수치제어 공작기계에 피가공물을 이체하는 용도의 시스템이 되도록 하였다. 본 연구를 위하여 한국과학원 기계공학파팀이 참가하여 몸체 및 핸드의 설계제작을 담당하였고, 제어연구실팀은 마이크로프로세서를 이용한 제어장치 및 써보모터의 구동장치를 설계 제작하여 하나의 시스템으로 규합하였다. 본 논문에서는 제작된 로봇트를 편의상 카이셈로봇트 1호(KAISEM ROBOT I)라

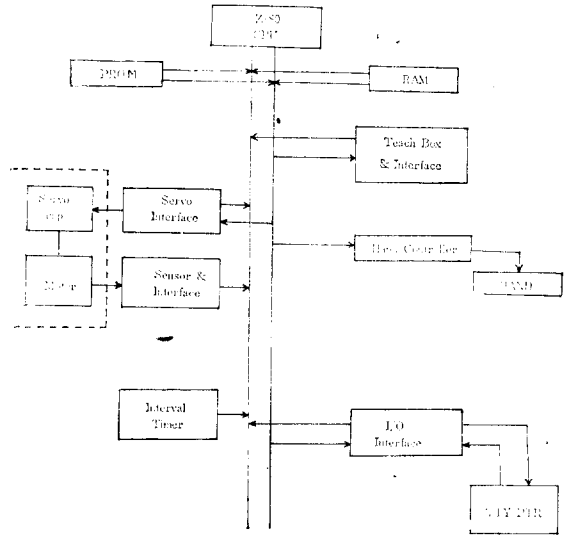
* 正會員: 韓國科學院 博士過程
** 正會員: 동아방송 기술개발실
*** 正會員: 韓國科學院 教授·博士
接受日字 1980年 5月 26日

1) LSI: Large scale integrated circuit.

고 명명하다. 카이젠로봇 1호는 크게 조작부(manipulatory unit), 제어부(control unit), 동력부(power unit)로 구성되어 있다. 몸체, 핸드 및 구동장치로 이루어진 조작부는 카이젠로봇 1호의 기계적인 외형을 구성하며. 로봇몸체의 움직임은 원통좌표방식에 준해서 R-축, θ -축, Z-축방향의 3개의 자유도와 핸드에서의 포획운동, 회전운동의 2개의 자유도를 갖고 있고, 각각의 구동부분은 고성능 DC 서어보모터 3대와 2조의 슬레노이드밸브로 작동하는 실린더로 이루어져 있다. 제어부는 로봇의 위치 및 속도를 제어하고, 조작부가 원하는 작업에 준해서 운동할 수 있도록 구동부에 적절한 제어명령신호를 인가하고 조작부의 운동을 감시하며 일체의 비상순간에 대비한 여러가지의 제어기능을 갖추고 있다. 카이젠로봇 1호의 제어은 표 1-1과 같으며 그림 1-2는 제작된 카이젠로봇 1호가 피가공물을 운반하는 실물도를 보이고 있다.

로봇의 제어장치하드웨어는 그림 1-1과 같은 블록도로 표시되며 중요한 하드웨어는 다음과 같다.

조작부가 구동할 수 있도록 구동부에 공급할 동력으로서 몸체는 DC 서어보모터를 구동하기 위해서 전기식동력을, 핸드는 공기식동력을 사용하였는바, 이들 서어보장치 및 핸드제어장치는 참고문헌 (18) 을 참조



- (i) Z-80 μ P를 CPU로 하는 μ -Computer Control system.
- (ii) Teach-Box 및 Interface.
- (iii) Servo-unit 및 Interface.
- (iv) Hand-controller.
- (v) TTY/PTR(Paper-Tape Reader)과의 Intefrace.

그림 1-1. Block diagram of control hardware

표 1-1. Specification

External frame	Cylindrical-Coordinate type		
Motion unit	3-DC Servo-motor, 0.8 kW/ea		
Power source	4KVA, AC 220V, 3 ϕ , 60Hz		
Motion range of axis	arm	z-axis (UP/Down)	950 \leftarrow \rightarrow 1,450mm (50mm)
		θ -axis (Rotation)	-105 $^{\circ}$ \leftarrow \rightarrow \pm 105 $^{\circ}$ (210 $^{\circ}$)
		R-axis (out/In)	1,044 \leftarrow \rightarrow 1,844mm (800mm)
	wrist	α -axis (Rotation)	0 $^{\circ}$ \leftarrow \rightarrow 90 $^{\circ}$ (90 $^{\circ}$)
Motion speed of axis	arm	Z-axis (Up/Down)	Max. 300mm/sec
		θ -axis (Rotation)	Max. 60 $^{\circ}$ /sec
		R-axis (Out/In)	Max. 300m/sec
	wrist	α -axis (Rotation)	60 \sim 90 $^{\circ}$ /sec (fixed)
Weight Capacity	maximum 30kg		
Control Scheme	PTP control, playback with memory (basic 80 points)		
Repeatability Accuracy	\pm 1.0mm		
Driving method	Move one axis at a time		
Application	Loading and unloading for a CNC machine tool		

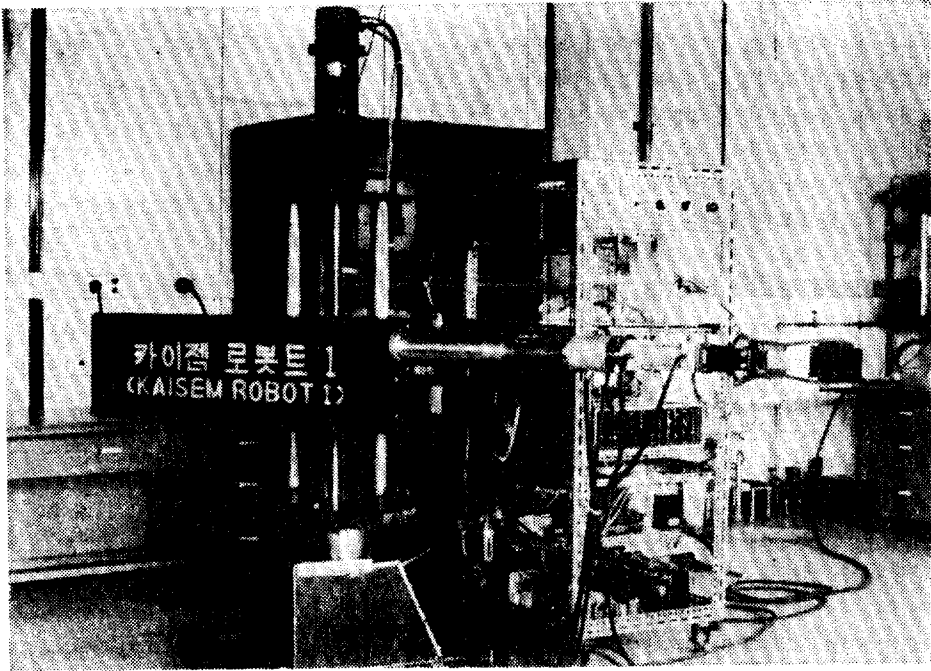


그림 1-2. 카이젬 로봇트 1호의 외관

하고 그의 (i), (ii), (v)항만 본 논문에서 기술한다.

2. 디지털 제어장치의 설계

제어부는 운전자로부터 작업동작명령을 받아 이로부터 실제작업수행에 필요한 제어신호를 발생시키기까지의 일체의 기능을 담당하며, 이를 하드웨어와 소프트웨어로 나누어 설명하기로 한다.

2.1 제어장치 하드웨어의 구성

하드웨어는 Teach-Box와 Servo 및 Sensor와의 Interface가 중요한 구성요소이다.

(1) Teach-Box

설계된 Teach-Box는 그림 2-1과 같은 외관을 갖고 있다. 윗부분의 A, B, C, D, E, F로 표시된 것은 7-Segment LED로서, A, B는 교시중에 있는 Memory block(후술함)의 주소를 표시하고, C는 한 Memory block 내의 Section number(후술함)를, E, F는 그 Section에 저장된 Data를, D는 프로그램중의 오차 및 로봇트운전중의 비상시 그 원인을 표시해 준다. 다음에 ①, ②, ..., ⑩으로 표시된 것은

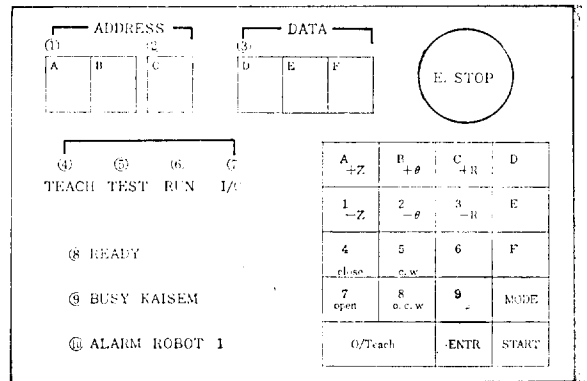


그림 2-1 Teach Box

LED-lamp로서 ①, ②, ③은 각각 Memory block, Section, Data 등을 입력시킬 차례가 되었을 때 켜진다. ④, ⑤, ⑥, ⑦은 각 LED-lamp 아래 표시된 Teach(교시), Test, Run, I/O 등의 Mode를 나타내며 Mode-Key에 의해서 한개의 LED-lamp가 켜진다. Key-board에는, 0, 1, 2, ..., 9, A, ..., F의 16개의 key와

MODE, ENTR, STATT의 3개의 key가 있는데, 이 key들 중에서 2중(重)으로 표시된 것들은 두 가지의 기능을 갖고 있다. 즉, 현재의 Mode와 LED-lamp ①, ②, ③의 점멸 상태에 따라 한 기능이 결정되는데, ①, ②, ③이 켜지면 각 key는 숫자로 작용하고 Teach-mode에서 ①, ②, ③이 모두 꺼져 있거나 로봇트가 원점에 없어서 START가 되기전의 상태에서는 A, B, C, 1, 2, 3 key는 각각 +z, +θ, +R, -z, -θ, -R 등의 "Jog-button"으로 작용한다. Jog-motion 후에는 ϕ-key는 Teach-key로 작용해서 눌렀을 때 로봇트의 현재의 각축의 위치를 지정된 Memory block에 기억시킨다. 4, 5, 7, 8의 4개의 key는 실제로 두 가지 기능을 갖고 있지 않고, 교시중에 S-code(후출함)로서 각각 ϕ4, ϕ5, ϕ7, ϕ8을 입력시키면 각 Key에 표시된 Hand-close, Hand-rotate c.w., Hand-open, Hand-rotate c.c.w.의 역할을 하도록 하였다. ENTR-key는 교시 및 Test중의 Data를 입력시키는데 사용된다. E-STOP button은 위급할 때 비상정지할 경우에 사용된다.

Teach-Box의 내부구조는 (i) Hex-encoded keyboard, (ii) Data의 입력 및 표시장치, (iii) Data의 출력장치로 구성되어 있다. 그림 2-2, 2-3, 2-4는 각각 이들의 실제회로를 표시한다.

(2) Servo 및 Sensor와의 Interface.

Servo 및 Sensor와의 Interface는 I/O port decoder, D/A converter, 리졸버신호처리회로등으로 구성되어 있다. 본란에서는 리졸버신호처리회로와 위치검

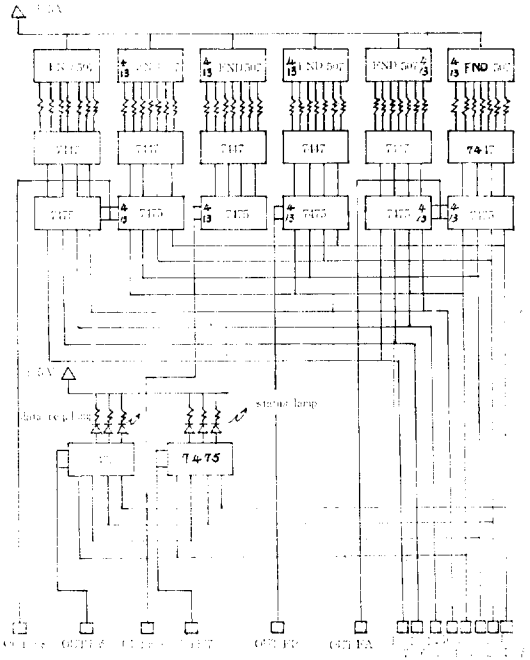


그림 2-3. Data receive part of teach box

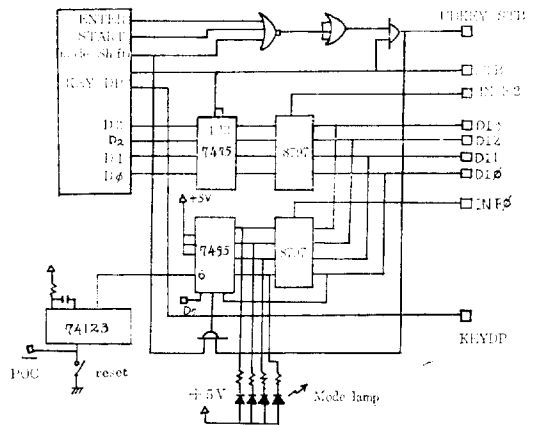


그림 2-4. Data sending part of teach box.

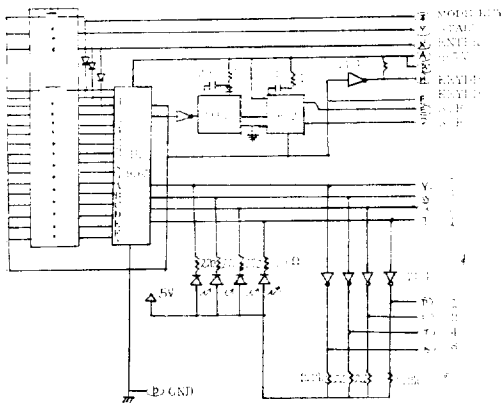


그림 2-2. Hex encoded key board circuit

출회로만을 설명하고, 다른 항목은 일반적인 것이므로 언급하지 않기로 한다.

리졸버는 정밀한 위치검출장치로서 2개의 회전자런선에 각각 $\sin\omega t$, $\cos\omega t$ 의 기준입력을 가하면 고정자

(2) I/O: Paper-tape reader, CRT-terminal과의 Input/Output operation

(3) 원점 : Origin, 기억부의 모든 위치 Data는 외부에 기계적으로 고정시킨 원점에 대한 상대 위치이다.

(4) Jog-button: Jog-button을 누르고 있는 동안만 로봇트가 운동한다.

권선에 유기된 출력은 $\sin(\omega t + \theta)$ 로 얻어지며 이때 θ 는 고정자와 회전자가 이루고 있는 기계적인 각도, 즉 모터축의 현재의 변위를 표시한다. 따라서 기준 입력의 위상과 출력신호의 위상차는 현재의 모터축의 위치가 되며 그림 2-5와 같이 위치 및 회전수를 검출한다.

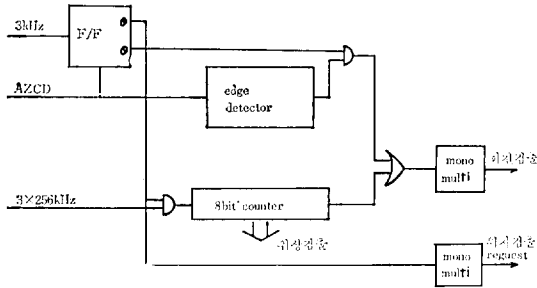


그림 2-5. 리졸버 신호처리 블록도

그림 2-5에서 AZCD 신호는 출력신호의 zero-crossing point를 검출한 구형파이며 한 회전을 256개의 펄스로서 카운트한다. 회전수검출방법은 C.C.W.운동시 즉 $\theta > 0$ 에는 edgedetector의 출력과 플립플롭의 출력 \bar{Q} 를 AND 함으로써 1회전당 1개의 pulse가 나오며, C.W.운동시(즉, $\theta < 0$)에는 8-bit 카운타의 carry로서 회전수를 읽는다. 검출된 회전수를 나타내는 펄스는 인터럽트를 이용하여 buffer의 내용을 변화시켜 준다. 한편 위상차의 검출은 플립플롭의 출력 Q의 falling-edge에서 인터럽트를 걸어서 8-bit 카운타의 내용을 μ -computer가 읽도록 하여 이루어진다. 최종단의 모노스테이블멀티바이브레이타(74123)은 모터의 회전속도가 낮을 때 여러번의 회전검출펄스가 발생함을 방지한다.

2. 2. 제어장치 소프트웨어

일반적으로 산업용로봇의 제어부 소프트웨어는 Hierarchical control scheme⁷⁾을 채용하여, 최상단의 level에서는 사람에게서 명령을 받아서 현재의 상태를 고려해서 작업의 방향을 결정하고, 두번째 level에서는 주어진 작업을 수행하는 신호를 만들어 내며, 맨 밑의 level에서는 위의 신호에 따라 실제작업동작을 제어하도록 되어 있다. 본 카이젠로봇 1호를 위하여 구성한 소프트웨어는, 최상단의 level에서, 운전자와의 교신에 사용되며 Teach-mode, Test-mode, Run-mode, I/O-mode 등의 4개의 Mode로 구성되어 있다.

(1) Teach-mode

Teach(교시: 敎示)는 로봇트 제어장치의 가장 중요한 부분으로서, 로봇트의 기억부에 (i). 위치(R, θ, Z), (ii). 속도, (7단계), (iii). 핸드의 회전 및 피가공물의 포획동에 필요한 S-code 등을 기억시켜 주는 일을 맡는다. 따라서 교시는 (i), (ii), (iii)의 data들로 이루어진 Memory block 단위(그림 2-6 참조)로 행하여진다.

Section	Contents
ϕ	speed
1	revolution of z axis
2	phase of z resolver
3	rev. of θ axis
4	phase of θ axis
5	rev. of R axis
6	phase of R resolver
7	s-code 1
8	s-code 2
9	s-code 3
A	s-code 4
B	s-code 5

그림 2-6. Teach block의 구조

그림 2-7, 2-8은 power-on 후부터 Mode 선택과 Teach-mode의 수행을 위한 소프트웨어의 flow-chart이다. 즉, 교시하기 위해서는 power-on 후 로봇트를 Jog-button($\pm Z, \pm O, \pm R$)으로 고정된 원점으로 보내면 Teach-Box에는 D를 제외한 모든 7-segment LED에 "0"이 표시된다. 이때 START-key를 누르면 교시를 시작할 수 있게 된다(이때 Ready-lamp가 꺼지고 Busy-lamp가 켜진다). 교시가 시작되면 Memory block의 주소를 나타내는 7-segment LED A, B에 첫번째 Memory block을 표시하는 $\phi 1$ 이, Section-number를 나타내는 LED C에는 ϕ (Memory block의 첫번째 section)이 표시된다. 맨처음 $\phi \dots 7$ 중의 적절한 속도 data를 입력시키기 위해서 숫자-key를 누른 후에 ENTR-Key를 누르면 속도 data가 첫번째 section에 기억되고 LED-lamp③은 꺼진다. 다음에 Jog-button을 눌러서 로봇트를 원하는 곳까지 운전한 다음 Teach-key를 누르면 각축당 2-byte씩 6-byte의 위치 data가 ①, ②, ..., ⑥번째 section에 기억된다. 이때 section-number를 표시하는 LED C는 7(8번째 section)로 바뀌고 LED-lamp③이 켜진다. 이때 최고 5개까지의 S-code를 ENTR-key를 사

5) Sensor overtravel, overload, overheat 등의 검출부.

용하여 입력시키면 5개 미만일 경우에는 " $\phi\phi$ "를 입력시켜 S-code end를 달린다. 이렇게 해서 한 Memory block의 교시가 끝나면 Memory block의 주소(A,B)는 하나 증가하여 $\phi 2$ 가 되고 section-number(C)는 다시 ϕ 이 된다. 첫번째와 같은 방법으로 교시를 계속하며, 최종적으로 교시가 끝나면 맨 마지막 block의 S-code에 "FF"를 입력시킴으로서 교시가 완료된다.

(2) Test-mode

교시된 내용의 점검 및 수정에 필요한 Mode로서 그림 2-9의 Test-mode에 필요한 소프트웨어의 flow-chart를 참고하면, 교시가 끝난 후 Mode를 Test로 바꾸고 START-key를 누르면 LED-lamp①(A,B상단)이 켜진다. 이때 Test하고자 하는 Memoryblock의 주소를 입력시키면 LED-lamp②가 켜지고, 원하는 section-number를 입력시켜주면 LED E,F에 교시된 data가 나타나면서 LED-lamp③이 켜진다. 내용을

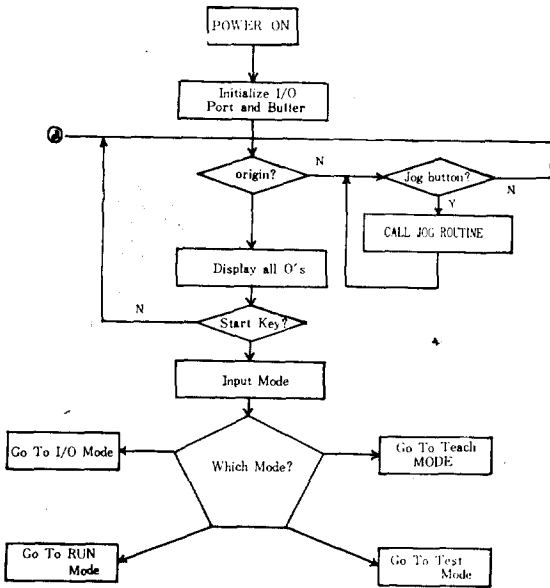


그림 2-7. 모우드 선택 flow chart

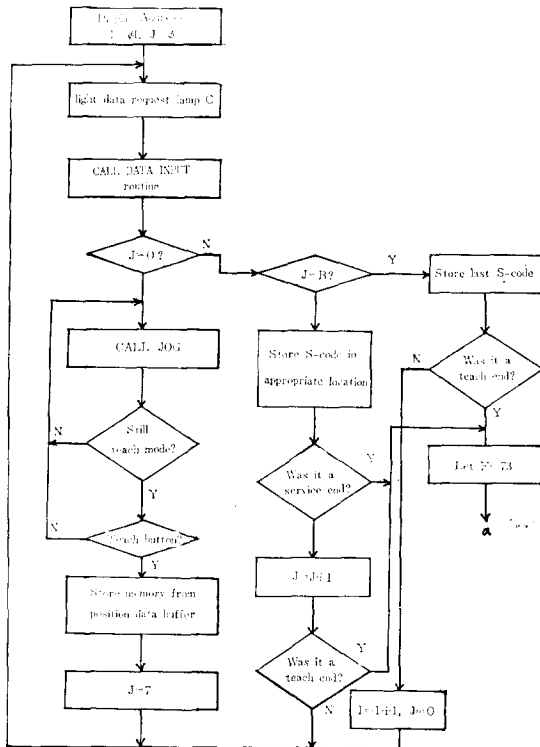


그림 2-8. Teach Mode

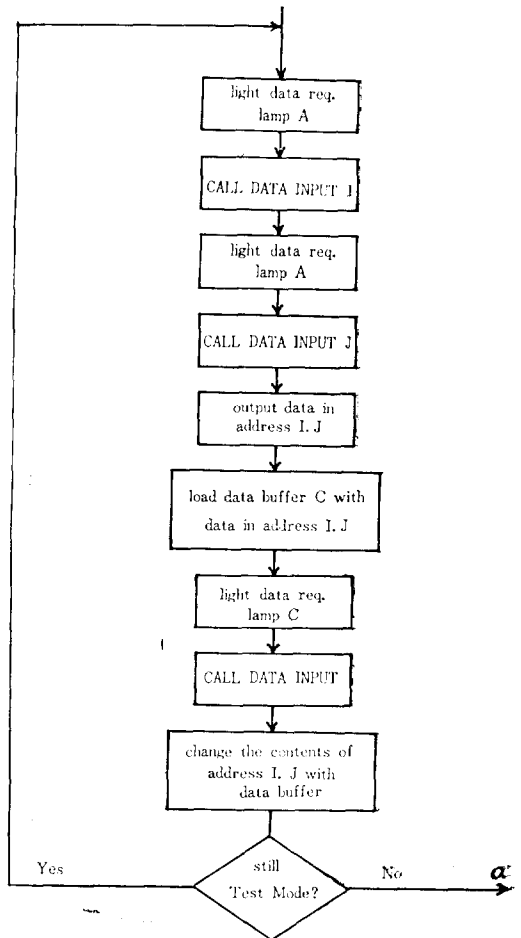


그림 2-9. Test mode

바꾸려면 새로운 data를 입력시키고, 그렇지 않으면 그냥 ENTR-key를 누른다. 이때 다시 LED-lamp ①이 켜지면서 앞의 과정을 반복한다. Test가 끝난 후에는 MODE-key를 눌러서 다른 Mode로 바꾼 후 ENTR-key를 누르면 된다.

(3) Run-mode.

교시된 프로그램의 Test가 끝난 후에 로봇트를 playback(재생작동)하기 위해서 사용되며, 그림 2-10, 2-11, 2-12의 flow-chart에 준해서 playback이 수행된다. 먼저 Run-mode를 선택한 후 START-key를 누르면 교시된 순서(Memory block 순서)대로 동

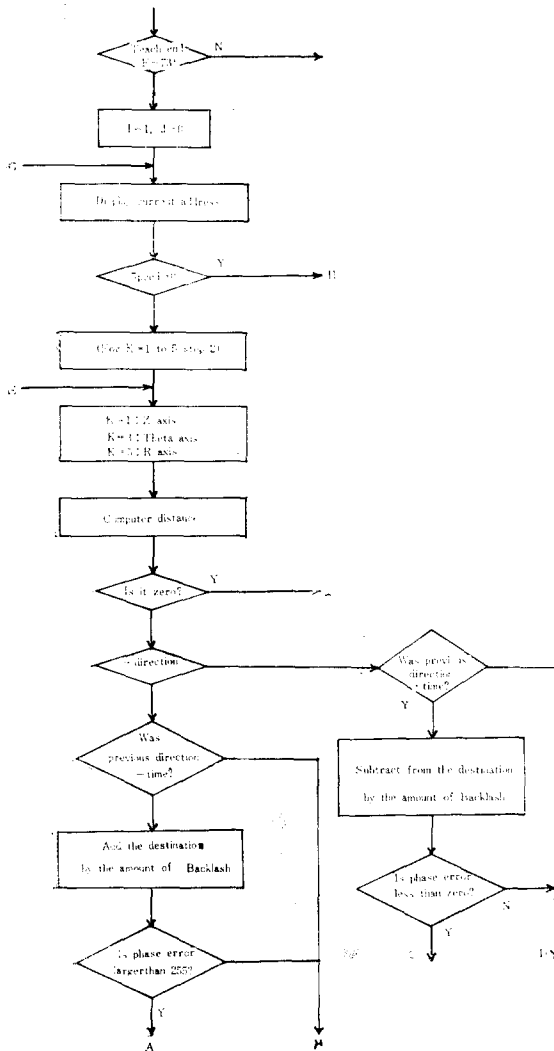


그림 2-10. Run mode flow chart (1)

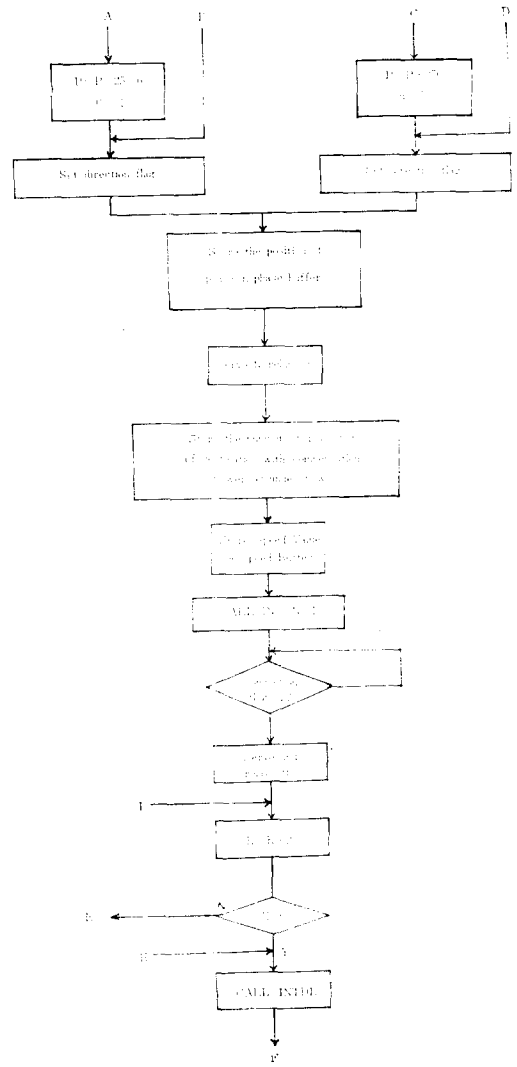


그림 2-11. Run mode flow chart (2)

작이 진행되며 이때 Teach-Box에는 현재 수행중인 Memory block의 주소가 LED A,B에 나타난다. 만일 위급한 상황이 발생하였거나 기타의 오동작이 생기면 즉시 E-STOP-button을 눌러면 로봇트가 급히 정지된다. E-STOP-button이 눌러지면 status-lamp ⑥, ⑨, ⑩ 중의 alarm-lamp ⑩이 켜지고 LED D에 error-code가 표시된다. E-STOP-button에 의해 정지한 로봇트를 다시 운전시키려면, control-panel 참고문헌 (18)참조 의 Release 스위치를 ON시키고 control-panel의 운전-mode(Auto, manual)를 manual

6) error-code; emergency, overtravel, overload, overheat 등에 의한 급정지에 표시된다.

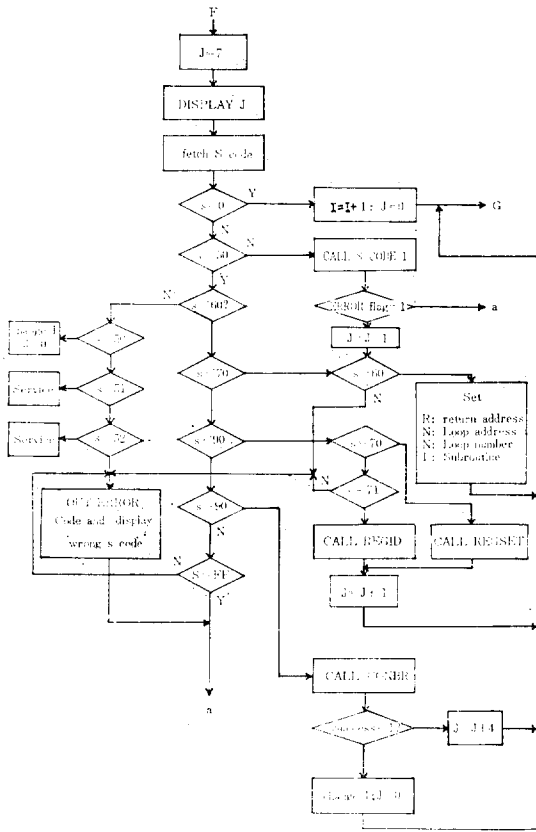


그림 2-12. Run mode flow chart (3)

로 바꾸고 로봇트를 Jog-button으로 원점까지 보낸 후 Release 스위치를 OFF 하고 운전-mode를 Auto로 전환한다. LED에 모두 "0"이 나타나면 교시부터 다

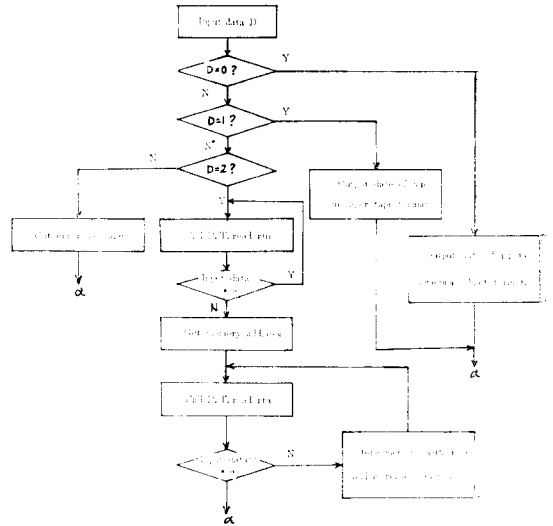


그림 2-13. I/O mode flow chart

시 시작한다.

(4) I/O-mode

한번 교시된 내용을 반복사용할 때에 프로그램을 paper-tape로 뽑아 놓거나 리스트를 만들어 놓고자 할 때에 사용한다. 프로그램이 완전하다고 판명된 다음에 I/O-mode를 선택하고 paper-tape를 뽑기 위해서는 "0"을 리스트를 만들고자 할 경우엔 "1"을, 뽑아 놓은 paper-tape를 load 시키고자 할 때는 "2"를 입력시키며, 그림 2-13은 I/O-mode를 위한 flow-chart이다.

표 2-1. Scode Table

SCode	Service	SCode	Service
00	S-Code end	50	Branch (1)
01	return (Subroutin end)	51	Wait check (1)
02	(Spare)	52	Dwell (1)
03	"	:	.
04	hand Close	:	.
05	hand rotate C.W	60	Sub program (1), (2)
06	.	:	.
07	hand open	:	.
08	hand rotate C.C.W	70	register set (1), (2)
09	.	71	register INC/DEC (1), (2)
0A	.	:	.
:	.	90	Conditional branch (1), (2), (3)
:	.	FF	Teach END

(5) S-code.

로봇의 동작중 translation, rotation 이외의 모든 동작은 S-code에 의해 수행된다. S-code는 크게 2가지로 나누어진다. 즉, 그 자체로서 수행가능한 Simple S-code와 다른 Memory block의 data가 필요한 Extended S-code가 있으며, 표 2-1에서 보는 바와 같이 왼편의 것들은 전자에 오른편의 것들은 후자에 포함된다.

3. 예 제

본 절에서는 전술한 제어부의 소프트웨어를 이용하여 예로서 간단한 작업의 실제 프로그램을 설명하기로 한다. 즉, 그림 2-14와 같이 콘베이어 C를 통해서 두 종류의 물건(a, b)이 점 P₂의 위치에 교대로 a, b, a, b, a, ...로 운반되어 올때, 물건 a는 5개씩 상자 A에

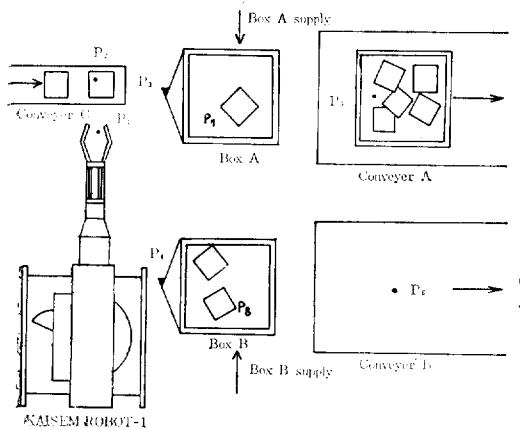


그림 2-14. Example of working station.

담아서 콘베이어 A에 실어보내고, 물건 b는 16개씩 상자 B에 담아서 콘베이어 B에 실어 보내는 작업을 32회 반복하는 경우를 생각해 보자, 단, 이때 상자 A,

Z Q R

P₁; (1 φ 67, φ 523, φ 32 φ)

P₂; (1 φ 67, φ 523, φ 789)

P₃; (φφBB, 1 φ 99, 23 φφ)

P₄; (φφBB, 36 BC, φ 999)

P₅; (21 AA, 36 BC, A φ 23)

P₆; (21 AA, 2 φ 11, A φ 23)

P₇; (1 φ 67, 1355, 25 A φ)

P₈; (1 φ 67, 36 BC, φ 777)

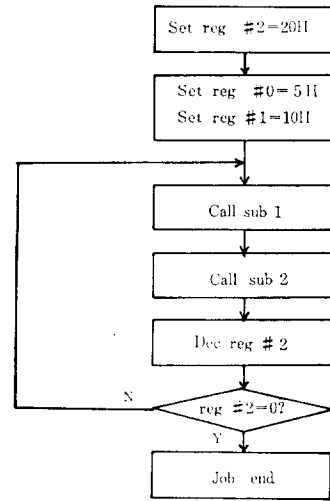


그림 2-15. Teach example

Kaisei Robot-1 Teaching Sheet

Address	Speed	Z-Position	T-Position	R-Position	S-code
1	00	00	00	00	70 02 20 70 00
2	00	00	00	00	05 70 01 10 00
3	00	00	00	00	60 09 01 00
4	00	00	00	00	60 15 01 00
5	00	00	00	00	71 02 00 90 02
6	00	00	00	00	00 13 50 03 00
7	05	10 67	05 23	03 20	51 01 07 00

그림 2-16. a.

7) 모든 숫자는 Hexadecimal number이며 앞의 두자리는 원점으로부터 모터의 회전수를, 뒤의 두자리는 모터축의 기계적 위상을 나타냄.

8	05	10 67	05 23	07 89	04 00
9	05	10 67	13 55	25 AO	07 71 00 00
A	00	00	00	00	90 00 00 12 01
B	05	00 BB	10 99	23 00	04 00
C	05	21	36	09	07 70 00 05 00
D	05	10 67	05 23	03 20	51 01 07 00
E	05	10 67	05 23	07 89	04 00
F	05	10 67	36 BC	07 77	07 71 01 00 90
10	00	00	00	00	01 00 18 01 00
11	05	00 BB	36 BC	01 EF	04 00
12	05	21 AA	20 BC	09 99	07 70 01 10 01
13	00	00	00	00	FF

그림 2-16. b.

B는 다른 장치에 의해 공급된다고 가정한다. 작업의 위치를 다음과 같이 잡자⁷⁾.

이러한 작업을 수행하기 위해서는 2개의 부프로그램이 필요하며, 부프로그램 1은 P_2 위치의 물건을 상자 A에 담고 5개가 됐을 때 상자 A를 콘베이어 A에 올려 놓은 후 다시 카운타를 5로 set 한다. 부프로그램 2는 P_2 의 물건을 상자 B에 담고 16개가 됐을 때 상자 B를 콘베이어 B에 올려 놓은 후 다시 카운타를 16으로 set 한다. 각각의 부프로그램을 32회 수행함으로써 작업이 끝난다. 작업을 위한 flow-chart, Teaching-sheet는 그림 2-15, 표 2-2에 보인 바와 같다.

4. 결 론

원통좌표형 산업용로봇을 playback-type으로 제어하기 위하여 μ -computer를 중심으로 소프트웨어 및 하드웨어를 설계하여 시험한 결과를 수록하였다. Command-level의 프로그램은 고급언어인 Basic Language를, Low-level에서는 Assembly Language를 사용하여 Memory space 문제와 프로그램의 수행 속도문제를 해결하였다. 그러나 본 시스템의 실용화를 위해서는 작업용프로그램언어를 개발하여 사용자가 쉽게 프로그램할 수 있게 하고, S-Code의 다양화를 기하여 여러 종류의 작업을 할 수 있도록 개선되어야 한다.

參 考 文 獻

1. E. Freud, M. Syrbe; "Control of industrial robots by means of μ -processors." JACC, 1978, p.167.
2. S.N. Saridis, C.S.G. Lee; "Heuristic control in trainable manipulators," JACC, 1976, pp. 712~716
3. G.N. Saridis, C.S.G. Lee; "An approximation theory of optimal control for trainable manipulators," IEEE Trans. on SMAC, 1979, March.
4. Shyh J. Wang; "A reprogrammable Industrial Robot control system," IEEE Trans. on SMAC, 1976, May.
5. R. Hariharan, et al; "Design of a μp -based numerical control system," IEEE Trans. on IECI Vol. 25, No. 4, 1978, Nov. p.355.
6. Tadashi Nagata, et al; "Robot planning system based on problem solvers", Electro-technical Lab. Tokyo, Japan.
7. NBS special publication 500-23; "An architecture for a Robot Hierarchical control system."
8. 로봇의 標準化 に関する 調査研究 報告書(第一報) 日本産業用 로봇工業會.
9. 7-th international symposium on industrial robots, 19-21, Oct. 1977, Tokyo, Japan.
10. 변중남; "산업용로봇의 제어", 공업기술 제 1권 3호, 1979, 6월.
11. 8-th international symposium on industrial

- robots, IPA, Stuttgart, West Germany.
12. Richard Paul; "Robots, models, and automation", Computer, July, 1979, pp.19~27.
 13. George N. Saridis; "Towards the realization of Intelligent controls," Proc. of IEEE. Vol. 67. No. 8. Aug. 1979.
 14. Mostek. Technical manual (z-80 cpu)
 15. Cromemco single card computer instruction manual.
 16. FANUC robot model 1 operating manual.
 17. 박영제; "산업용로봇트 몸체의 설계 및 제작", 한국과학기술원 석사학위논문, 1980년 1월.
 18. 송상섭; "산업용로봇트의 서어브제어장치에 관한 연구", 한국과학기술원 석사학위논문, 1980년 1월.
 19. Arthur Astrop; "Much to learn about robots," Machinery and production eng. 22, March, 1978.
 20. Arthur Astrop; "Robots need more help than you think", Machinery and production Eng. 6. Dec., 1978.
 21. Kanji Yonemoto, JIRA; "Will the industrial robots change the course of history?", Journal of Electronic Eng. Feb, 1975.