

Micro Computer 에 의한 機械 · 裝置의 制御方法 ⑦

5. 스테핑모터使用技法(계속)

(4) 驅動用 프로그램例(2) (스테핑모터增速, 減速 驅動用 프로그램)

前月號에서 설명한 바와 같이 스테핑모터는 回轉數를 정하는데 0.9度라든지 1.8도 단위의 精度로 회전시키는 角度를 결정할 수 있을 뿐만 아니라 그 회전하는 속도도 정확히 제어할 수 있는 특징을 가지고 있다.

즉 回轉은 예를 들면 32회와 12/400회만 회전시키는 요구에도 견딜 수 있으며 또한 그 회전속도를 작은 속도로도 결정할 수 있다는 것이다.

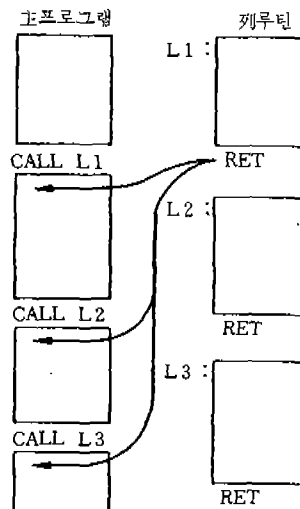
따라서 그 회전속도를 자유자재로 결정할 수 있다는 것을 이용하여 스테핑모터를 점차로 빨리 구동시키는 「增速驅動」과 그 반대로 고속도로 회전하고 있는 상태에서 점차로 減速시켜 정지시키는 「減速驅動」을 할 수 있다는 것을 알게 된다.

따라서 간단한 예로서 ① 먼저 低速度로 始動시키고 ② 다음에 中速度로 올리고 ③ 이어 희망하는 回轉數만큼 高速度의 일정한 속도로 回轉시키고 ④ 다음에는 中速度로 내리고 ⑤ 끝으로 低速度로 내려 ⑥ 所定の 회전위치에서 정지시키는 방법이 있다. 간단하면서도 일반적인 경우에는 충분하므로 많이 사용되고 있다.

여기서 설명하는 이 增速, 減速驅動은 어느 정도

중량이 있는 물체를 갑자기 어떤 속도로 始動시키는 것은 無理이며 또한 정지시키는 것도 갑자기 정지시키면 좋지 않다는 것은 電車 등에서와 같이 당연한 것인데 그러나 반면에 가급적 빨리 作業을 시키려고 하므로 일정한 속도의 高速驅動의 고속의 정도는 가급적 빠른 속도를 사용하려고 하는 요구가 있기 때문에 스테핑모터 사용의 경우에 많이 필요로 하는 技法이다.

이 경우뿐만 아니라 컴퓨터의 최대의 強點은 다수의 「각종의 作業을 하기 위한 각각의 프로그램」을 독립된 列루틴으로서 메모리 속에 몇組든지 그림 41과 같이 루프 No. 1을 위한 것(라벨 L1), 루

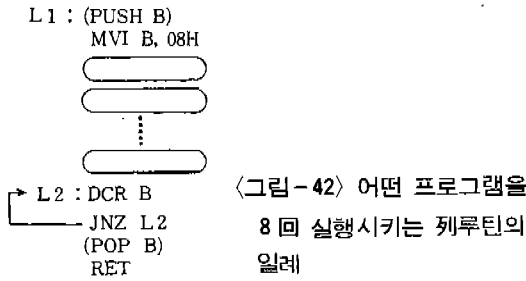


〈그림-41〉 列루틴의 호출방법과 복귀방법

프 No. 2를 위한 것(라벨 L2), 루프 No. 3를...와 같이 대비시켜 놓고 **主 프로그램**에 의하여 이들을 L1 (가령 **低速度驅動**을 위한 프로그램)을 1회, L2 (**中速度驅動**을 위한 프로그램)을 1회, L3 (**高速度驅動**을 위한 프로그램)을 16회, 그리고 L2를 1회, L1을 1회 실행하고 **驅動**을 정지하는 명령을 발하도록 하면 **스태핑모터**는 점차로 **増速**하여 고속으로 16회 회전하고 점차로 **減速**하여 合計 20회를 회전하고 정지하게 된다.

간단한 **自動式**의 로더나 언로더, **産業用** 로보트 등을 자유자재로 동작시키는 것도 대부분 이같은 **技法**의 **應用**과 **擴張**으로서 이들을 위해서는 **多段階方式**의 **加速**, **減速驅動**에서 사용되는 더욱 발전된 **技法**를 병용한다.

여기서 이들의 경우에 常用되는 하나의 프로그램 법에 대하여 복습하기로 한다. 그것은 예를 들면 라벨이 L1의 **레루틴**을 8회 실행시킬 때에는 **그림 42**와 같이 먼저 가령 **B 레지스터**에 8을 **格納**해 놓고 **B 레지스터**의 내용을 변화시키고 싶지 않을 때에는 「Push B, Pop B」로 **전체**를 끼운 형태의 「그 작업을 위한 프로그램」을 배열하는 방법이 常用된다 (**B 레지스터**의 내용의 보존이 필요치 않을 때에는 괄호를 한 부분은 생략할 수 있다)



이 방법은 이것을 실행시키는 **回數**를 변화시키려면 이 **數值**의 8을 변화시키는 것만으로 256회 이하이면 같은 형태의 것으로 된다.

그러나 앞서와 같이 2회라든지 3회만을 실행할 경우에는 **레루틴** L1은 다만 이 「그 작업을 하는 프로그램」 뒤에 **RET**를 둔 것만으로 하고 **主 프로그램** 쪽에 다만 이를 실행하려는 회수의 **個數**만큼 「Call L1」을 배열하기만 하면 된다 (**그림 43**).

이것은 당연한데 때로는 이 방법을 잊거나 또는 이 방법이 발달된 방법인 2개의 「Call L1」의 위

CALL L1

CALL L1

(그림-43) **레루틴을 2회 실행시키는 간단한 방법**

에 가령 「DCRE」 등의 어느 레지스터의 내용을 변화시키는 명령을 배치하고 그 후의 **콜** 명령을 CC나 CM 등의 **조건부 콜** 명령으로 하고 2회까지는 반드시 콜하여 실행시키는데 그 뒤에는 만일 어떤 **條件**이 **充足**되었을 때에 콜하는데 **조건**이 충족되지 않으면 콜하지 않고 그 프로그램 앞의 **스텝**으로 진행시키는 **技法**를 활용할 수 있다는 것을 기억해두도록 한다.

또한 때로는 실패하는 수가 있으므로 이 **多種**의 **조건부 콜** 명령이 발해졌을 때 특별히 주의해야 될 것이 하나 있다.

그것은 **점프** 명령과 **콜** 명령의 차이도 포함이 되는데 **점프** 명령 (**조건부 점프** 명령을 포함시키면 8085용에 9종이 있다)에서는 가령 **그림 41**의 **主 프로그램**에서 **第 1번**의 「Call L1」을 **실수**로 **점프**의 명령으로 하면 (가령 **CD**를 **C3**으로 **入力**했을 때) **L1**의 **레루틴**을 **實行**한 후 **主루틴**으로는 돌아갈 수 없고 프로그램의 실행은 **暴走**해 버린다.

복귀시키는 방법은 **L1**의 최후의 「RET」 대신 「JMP」 복귀선의 **애드레스**와 「**레루틴** 중의 어떤 번지」를 **JMP** 뒤에 두게 된다. 그러나 **第 1번**의 「Call L1」이 **점프** 명령이 된 위치에서 루프 **L1**으로 왔을 때에도 **第 2번**의 「Call L1」의 번지에 놓인 **점프**의 **번호**로 루프 **L1**이 실행된 경우에도 동일한 번지로 복귀하게 된다.

이렇게 되는 것이 좋은 특별한 경우도 있는데 일반적인 경우에는 **第 1번**의 **콜** 명령, **第 2번**의 **콜** 명령, **第 3번**의... 등의 경우에도 루프 **L1**을 실행한 후에는 그들의 **콜** 명령 바로 다음의 번지로 복귀하려고 하는 경우가 많다.

이와 같은 목적을 위해 편리하게 되어 있는 것이 이 **콜**의 **번호**에서 **조건부 콜**을 포함하면 역시 9종류 정도의 모두가 **리턴** 명령을 사용할 수 있으며 또한 더욱 고급의 사용방법으로서 「**正**이면 복귀한다」든지 「**負**이면 복귀한다」 (당연히 그 **조건**이 충족되지 않으면 복귀되지 않고 그 **번호**를 무시하고 프로그램 앞의 **스텝**으로 진행한다)는 **조건부 리턴** 명령이 9종류 준비되어 있다 (루프 **L2**, 루프 **L3**를

이용하는 경우에도 마찬가지이다).

이로써 여기서 설명하려고 하는 「3段式増速, 定速, 減速驅動用 프로그램」을 위한 豫備設明이 끝났으므로 이를 실행시키기 위한 모든 프로그램을 그림 44에 들었다. 그러나 이것은 實驗結果 동작은 하는데 좀더 짧게 깨끗하게 작성하는 방법이 있으나 특별히 다소 길게 되었어도 알기 쉽게 되어 있다.

어드레스順으로 설명한다. 최초의 번지가 8100으로 되어 있는데 이것은 이 마이크로컴퓨터의 사용자가 사용할 수 있는 RAM部分이 8000부터 이므로 만일 이 프로그램 앞에 어떤 프로그램을 두고 싶은 경우가 뒤에 발생했을 때를 위해 임시로 선정한 것으로 가령 8020정도라도 충분한 것이다.

8100 3E 09	: 포트 포트	8213 00	
02 D3 03	: 設定	14 00	
04 3E 00	: 出力포트에	15 1E C8	: E에 C8H를
06 D3 02	: 00을	17 78	: B를 Acc에
08 00		18 D3 02	: C포트에 出力
09 C5	: B, C保存	1A CD 45 82	: 「中運用타이머」를 呼出
0A 06 05	: B에 05를	1D 79	: C를 Acc에
0C 0E 04	: C에 04를	1E D3 02	: C포트에 出力
0E CD 00	82: 「低速」을 實行	20 CD 45 82	: 「中運用타이머」를 呼出
11 CD 15 82	: 「中速」을 實行	23 10	: E를 -1
14 26 10	: H에 10H를	24 C2 17 82	: E가 0까지 8217로 부귀
16 00		27 C9	: 主루틴으로 부귀
17 00		28 00	
18 CD 2A 82	: 「高速」을 實行	29 00	
1B 25	: H를 -1	2A 1E C8	: E에 C8H를
1C C2 18 81	: H가 0까지 8118로 부귀	2C 78	: B를 Acc에
1F CD 15 82	: 「中速」을 實行	2D D3 02	: C포트에 出力
22 CD 00 82	: 「低速」을 實行	2F CD 4A 82	: 「高速運用타이머」를 呼出
25 CD 70 82	: 약 1.8초 대기	32 79	: C를 Acc에
28 C1	: B, C를 부귀시킨다.	33 D3 02	: C포트에 出力
29 00		35 CD 4A 82	: 「高速運用타이머」를 呼出
2A C5	: B, C를 保存	38 1D	: E를 -1
2B 06 01	: B에 01을	39 C2 2C 82	: E가 0까지 822C로 부귀
2D 0E 00	: C에 00을	3C C9	: 主루틴으로 부귀
2F CD 00 82	: 「低速」을 實行	8240 3E 00	: Acc에 00을
32 CD 15 82	: 「中速」을 實行	42 C3 4F 82	: 824F로 점프
35 00		45 3E AA	: Acc에 AAH를
36 00		47 C3 4F 82	: 824F로 점프
37 26 05	: H에 6을	4A 3E 60	: Acc에 60H를
39 CD 2A 82	: 「高速」을 實行	4C C3 4F 82	: 824F로 점프
3C 25	: H를 -1	4F 3D	: Acc를 -1
3D C2 39 81	: H가 0까지 8139로 부귀	50 C2 4F 82	: Acc가 0까지 824F로 부귀
40 CD 15 82	: 「中速」을 實行	53 C9	: 主루틴으로 부귀
43 CD 00 82	: 「低速」을 實行	70 C5	: B, C保存
46 C1	: B, C를 부귀시킨다	71 D6	: D, E保存
47 76	: 終了	72 16 05	
48 00		74 01 00 00	
8200 1E C8	: E에 C8H를	77 0B	
02 78	: B를 Acc에	78 78	
03 D3 02	: C포트에 出力	79 B1	約 1.8초의 타이머루틴
05 CD 40 82	: 「低運用타이머」를 呼出	7A C2 77 82	
08 79	: C를 Acc에	7D 15	
09 D3 C2	: C포트에 出力	7E C2 74 82	: D, E를 부귀시킨다.
0E CD 40 82	: 「低運用타이머」를 呼出	81 D1	: B, C를 부귀시킨다
0B 1D	: E를 -1	82 C1	: 主루틴으로 부귀
0F C2 02 82	: E가 0까지 8202로 부귀	83 C9	
12 C9	: 主루틴으로 부귀		

〈그림-44〉 스테핑모터의 増速, 減速 驅動用 프로그램

性能이 좋은 테이프레코더를 사용하여 프로그램을 테이프에 넣거나 테이프에서 마이크로컴퓨터의 RAM에 넣거나 하는 경우에는 문제가 없는데 경험 상 예를 들어 그림 44의 프로그램에서 8100번지부터 최후의 8283번지까지를 테이프에 넣었을 때 최초의 3E가 3D가 되거나 최후의 번지 내용이 정확치 않은 數値가 되어 들어가 버리는 것을 경험하고 있다. 그러나 이것을 방지하기 위해 8100番地の 하나 적은 번지인 80FF부터 최후의 번지의 하나 뒤의 번지까지를 넣도록 하면 전혀 이같은 염려가 없어진다. 또한 8108이나 8116등에 「00」의 NOP 命令의 機械語가 들어 있는데 이것은 현재는 「D3 02」로 프로그램되어 있는데 表示의 목적에서 이 뒤에 「D3 01」등을 삽입하고 싶은 때에 다수의 번지가 변화하게 되는 것을 방지하기 위해 있는 것이다. 이 경우에는 정리하여 배버러도 프로그램의 효과는 마찬가지이다.

먼저 전체적으로 보면 ① 8100에서 8148까지는 主루틴이고 이에 의하여 스테핑모터는 増速하면서 시동하여 正方向으로 어떤 回轉數만큼 회전하고 減速하면서 정지하며 (전체적으로 20회전을 하고) 잠시 정지한 후 마찬가지로 増速, 減速하면서 반대방향으로 10회전하고 정지하는 프로그램으로 되어 있다 (물론 이 回轉數나 동작의 변경은 프로그램의 의미를 알면 용이하게 할 수 있다). ② 8200에서 8212까지는 低速回轉을 위해 ③ 8215부터 8227까지는 中速回轉을 위해 ④ 822A부터 823C까지는 高速回轉을 위한 부분 ⑤ 8240부터는 低速回轉을 위한 타이머루틴 ⑥ 8245부터는 中速回轉을 위해 8244부터는 高速回轉을 위한 타이머루틴 ⑦ 8270부터 8283까지는 약 1.8초 정도의 타이머루틴이다.

지금까지 설명한 부분을 요약하면 8109와 8128에서는 다른 부분에서 사용된 레지스터의 내용을 보존하고 있다.

810A에서 B레지스터에 「05」를 넣고 810C에서 C레지스터에 「04」를 넣은 것은 C포트에 「05」를 出力하면 스테핑모터는 正轉으로 펄스의 「1」의 부분을 送出하며 「04」를 出力하면 正轉으로 「0」의 부분을 送出하므로 이 「05」와 「04」를 언제든지 반복하여 사용할 수 있도록 각각 B와 C의 레지스터에 格納한 것이다.

*