Micro Computer 에 의한 機械·裝置의 制御方法 ④

4. 각각의 타스크를 위한 프로그램

마이크로컴퓨터는 상상 이상으로 복잡한 각종의 制御를 高速으로 실행하는데 이를 위해서는 그것이 매우 복잡하고 긴 프로그램이라도 이것을 만들어 마이크로컴퓨터에 미리 格納해야 된다. 따라서 형식이 이미 정해져 있는 給與計算用이라든지 原價計算用이라든지, 게임에서는 인베이더게임用이나 野球게임用 등을 위한 프로그램은 이것을 작성하여 카세트테이프에 넣어 多數의 유명메이커가 發賣하고 있다. 프로그램의 內容을 변경하지 않고 사용하면 이것을 구입하여 마이크로컴퓨터에 넣어 사용하는 방법도 최근에 증가되고 있다.

그러나 이 경우에도 주의해야 되는 것은 일반적으로 많이 사용되는 BASIC言語에 限定시켜도 20種 정도가 있으며 각각의 퍼스널컴퓨터에서 사용할 수 있는 命令이 다르다. 여기서 정확히 자기가 사용하는 퍼스널컴퓨터의 명칭을 말하고 거기에 사용할 수 있는 「目的하는 프로그램」을 위한 카세트를 구입해야 된다는 주의가 필요하다.

그러나 設定値制御用이나 스테핑모터나 일반적인 電動機를 자유롭게 제어하여 回轉시키는 등의 企業의 機械나 장치를 제어하기 위한 프로그램은 그 때마다 変化하여 미리 만들어 둘 수가 없는 수가 있으며 이것은 사용하는 기술자 자신이 만들어 사용해야 된다.

따라서 적당히 연구하여 프로그램을 작성하게 되는데 상당히 긴 프로그램이라도 잘 살펴보면 이것은 多數個의 「個個의 작업을 하기 위한 프로그램(「타스크」를 위한 프로그램)」의 혼합으로 구성되고 있다는 것을 알 수 있다.

따라서 많이 필요로 하는 作業을 위한 프로그램을 가급적 많이 入手하여 이것을 보존해두고 필요한 때에 이들을 구성하여 사용한다. 이들을 構成하여 사용하는 데에도 여러가지의 技法이 있는데 이것도 定石과 같은 것으로 「常用의 技法」을 반복하여 사용하므로 한번 마스터하면 비교적 용이한 것이다.

따라서 지금까지 사용한 것이라든지 앞으로 필요해질 「타스크를 위한 프로그램」에 대하여 整理하기로 한다.

(1) 어떤 機器를 始動시키거나 停止시키거나 하는 프로그램

가장 많이 필요로 하는 것의 하나로 마이크로컴퓨터 내에서 판단한 결과 그같은 狀況이라면 어떤 電動機를 始動시킨다. 그 狀態에서는 정지시킨다는 指令을 내리는 것이다.

미리 出力포트를 가령 C포트로 設定하는 프로그램에서는 **그림17**과 같은 프로그램을 실행하면 된다

1行째와 2行째는 모드와 入出力포트를 設定한다. 3行째에서 01이 한번 Acc에 넣어져 이것이 P

```
MVI  A,90 H
OUT  03 H
MVI  A,01 H
OUT  02 H
```

〈그림－17〉 機器를 始動시키거나 停止시키거나 하는 프로그램

C에 出力된다. 이 예에서는 PC_0에만 1이 PC_1 ~ PC_7에는 0가 出力되므로 PC_0에 연결되어있는 機器가 ON으로 되고 다른 端子에 연결되어 있는 것은 ON이 되고 있어도 모두 OFF로 된다.

PC_1에 연결되어 있는 것은 ON으로 하기위해서는 3行째의 01을 02로, PC_2에 연결되어 있는 것을 ON으로 하려면 이것을 04로 하면 된다. (모두를 OFF로 하려면 00로 한다).

(2) 어느 리미트스위치 LS 나 푸시버튼스위치가 ON 으로 되었는지 여부를 檢出하는 프로그램

이것을 실행하는 것뿐이라면 前述한 그림18과 같은 프로그램이면 된다.

이것의 変形으로 어떤 일련의 프로그램에 의한 작업(어떤 타스크)을 반복하여 實行시키는데 리미트스위치 LS가 들어가면 이 반복을 중지하고 다른작업으로 이행하려고 하는 경우가 있다. 로봇機構用 등에서도 각 동작마다 이같은 機能이 要請된다.

이를 위해서는 그 타스크를 위한 서브루틴의 처음에 그림19와 같은 3행을 두면 된다. 하아드構成으로서는 미리 PA_0의 端子에 그를 위한 리미트스위치 LS가 ON이 되었을 때 5V가 유도되며 PA_1 ~PA_7의 端子는 0V를 유지하도록 하면 된다. (00 H는 A 포트의 番地).

1行째에서 A 포트에서의 入力데이터(이경우에는 이 리미트스위치 LS가 ON이 되었을 경우에는0000 0001)가 Acc에 들어가고 2行째에서 이 入力데이터와 01의 앤드를 취할 수 있으므로 리미트 스위치 LS가 ON에서는 Acc에 1이 들어가고 結果가 0이 아니므로 점프한다. 따라서 이 경우에는 3行째에서 최후의 行으로 뛰어 메인루틴으로 돌아가 다른 프로그램으로 진행된다.

```
AA: IN  00 H
    SUI 01 H
    JNZ AA
```

〈그림－18〉 스위치의 檢出用 프로그램

```
AA: IN  00 H
    ANI 01 H
    JNZ ZZ
    JMP AA
ZZ: RET
```

〈그림－19〉 스위치가 들어갈 때까지 반복하여 實行하는 프로그램

리미트스위치 LS가 아직 ON이 아니면 반대로 3行째에서 점프하지 않으므로 4行째부터 뒤의 스텝을 차례로 실행하여 이 루프를 또 實行하게되며 끝에서부터 2行째에서 뛰어 다시 최초의 行으로 돌아오며 이상의 것을 반복한다.

應用으로서 리미트스위치 LS가 多數 존재하여 그 뒤에 계속하여 No.2 LS가 들어올 때까지 No.2의 타스크를 위한 서브루틴을, No.3 LS가 들어올때까지 No.3의 타스크를 위한 서브루틴을 …… 반복하는 동작을 시킬 경우에는 No.2 LS가 들어오면 PA_1에만 5V가 오도록 구성하고 2行째의 01H를 02H No.3 LS는 PA_2에 연결하고 2行째의 01H를 04H가 되도록 변경하는 것만으로 어느 리미트 스위치 LS가 들어갔다는 것도 檢出할 수 있다.

이것을 적용하여 최초에 A 포트에서의 入力 데이터를 넣어 이것을 01H, 02H, 04H, 08H, …… 등의 數値와 비교하여 어느 리미트스위치 LS가 들어간 경우에는 어떤 작업을 시킨다는 것도 할 수 있는데 이 때에는 「ANI00(5)」이 아니고 「CH00(5)」와 「JE」의 구성을 사용하는 것이 좋다. 가령 그림20과 같은 프로그램이 된다.

```
AA: IN  00 H
    CPI 01 H
    JZ  BB
    CPI 02 H
    JZ  CC
```

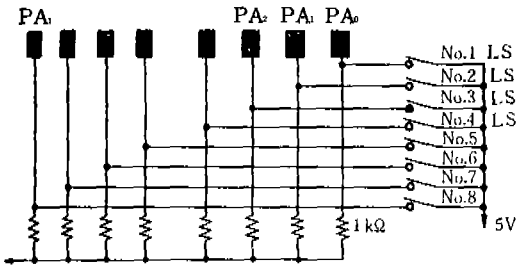
〈그림－20〉 多數의 스위치를 判斷하여 그 결과에 의하여 각각의 작업을 시키는 프로그램

ANI에서는 앤드를 취하여 그 결과를 Acc에 넣기 때문에 이 명령을 실행할때마다 Acc의 內容이 変化한다. 그러나 CPI命令에서는 比較를 위해 「(Acc)-數値」의 操作을 하는데 그 結果는 Acc에는 들어 갈 수 없으므로 몇번 이 命令을 실행해도 Acc에 들어간 데이터는 変化하지 않는다. 따라서 多數의 數値와 차례로 比較하여 그 結果를 보고 여러 방면으로 갈라져 점프하여 實行시키는 경우에는 편리하다.

이같은 比較命令에서는 Acc의 내용에서 그 경우의 数値가 차감되어 그것이 「0」이 되면 Z가 1이

되는 것을 이용하여 리미트스위치 LS가 들어간 경우에는 Z가 1이 되는 것을 사용하여 分岐시킨다.

이같은 프로그램에 의한 實驗을 해본다. 사실은 이 TK-85에서는 各 포트는 전부 보호를 위해 33KΩ의 抵抗体로 5V로 되어(풀업되어)있으므로 入力포트에 아무것도 연결하지 않고 앞으로의 데이터를 Acc에 넣으면 FFH가 들어온다. 여기서 프로그램을 알기 쉽게 하기 위해 이것을 전부 0으로 한다. 이를 위해 **그림21**과 같이 각 단자에 약 1kΩ정도의 抵抗에 의하여 TK-85의 GND에 접속해둔다. 그러면 各端子에는 약간의 電圧이 남아 있는데 마이크로컴퓨터의 성질상 0으로 判斷한다. 또한 리미트스위치 LS에서의 5V는 各端子에 직접 유도해 두면 리미트스위치 LS가 들어왔을 때에는 그 端子는 다시 5V가 된다.



〈그림-21〉 入力포트의 각각의 端子를 최초에는「0」으로 하는 方法

먼저「ANI」와「JNZ」의 구성의 실험을 하기 위해 **그림22**의 프로그램을 마이크로컴퓨터에 넣는다. 또한 RUN시켜보면 2進表示部의 1과 2의 리드가 약 0.3秒마다 교대로 點燈을 반복한다. 여기서PA0에 연결되어 있는 리미트스위치 LS를 ON으로 하면 1과 2는 소거되어 上位의 4개의 리드가 點燈되고 정지한다.

따라서「① 810B에서 81.9까지의 동작」을 반복하고 있으며 리미트스위치 LS가 들어가면 이것을 중지하고「② 811C에서 8120까지의 동작」의 실행으로 이행하는 것을 알 수 있으므로 이 ①과 ②를 자기가 希望하는 動作의 프로그램으로 변경하면 된다.

마찬가지의 리미트스위치 LS의 ON, OFF의 檢出은「RRC」와「JC」의 구성으로도 되므로 그 실험을 해본다. 이를 위해 **그림23**의 프로그램을 마이크로컴퓨터에 넣어 RUN시키면 앞의 경우와 마찬가지로 2進表示部에 1과 2의 表示를 반복한다. 여기서 이 例의 경우에는 (PA0가 아니고)PA1의 端子에 리미트스위치LS에 의하여 5V를 유도하면 앞에서와 마찬가지로 2進表示部의 上位의 리드 전부에 點燈되고 중지한다.

```
        MVI  A,90 H
        OUT  03 H
AA :    IN   00 H
        ANI  01 H
        JNZ  ZZ
        MVI  A,01 H
        OUT  01 H
        CALL TIM
        MVI  A,02
        OUT  01
        CALL TIM
        JMP  AA
ZZ :    MVI  A,F0
        OUT  01
        HLT
TIM :   LXI  B,0000 H
BB :    DCX  B
        MOV  A,C
        ORA  B
        JNZ  BB
        RET
```

〈그림-22〉 어떤 타스크를 리미트 스위치 LS가 들어올때까지 반복하는 動作을 ANI와 JNI JNZ로 실행시키는 테스트프로그램

```
8100  3E 90        MVI  A,90 H
  02  D3 03        OUT  03 H
  04  DB 00  AA :  IN   00 H
  06  0F           RRC
  07  0F           RRC
  08  DA 1C 81     JC   ZZ
  0B  3E 01        MVI  A,01 H
  0D  D3 01        OUT  01 H
  0F  CD 30 81     CALL TIM
  12  3E 02        MVI  A,02 H
  14  D3 01        OUT  01 H
  16  CD 30 81     CALL TIM
  19  C3 04 81     JMP  AA
  1C  3E F0  ZZ :  MVI  A,F0 H
  1E  D3 01        OUT  01 H
  20  76           HLT
  30  01 00 00 TIM :  LXI  B,0000 H
  33  0B       BB :  DCX  B
  34  79           MOV  A,C
  35  B0           ORA  B
  36  C2 33 81     JNZ  BB
  37  C9           RET
```

```
8100  3E 90
  02  D3 03
  04  DB 00
  06  E6 01
  08  C2 1C 81
  0B  3E 01
  0D  D3 01
  0F  CD 30 81
  12  3E 02
  14  D3 01
  16  CD 30 81
  19  C3 04 81
  1C  3E F0
  1E  D3 01
  20  76
8130  01 00 00
  33  0B
  34  79
  35  B0
  36  C2 33 81
  39  C9
```

〈그림-23〉 어떤 타스크를 리미트스위치LS가 들어올때까지 반복하는 動作을 RRC와 JC로 실행시키는 테스트프로그램

RRC의 命令은 Acc의 내용을 각각 1비트 만큼 우측으로 回轉시키고 있으므로 最低비트에서 2번째(PA1의 入力)가 1이 되었을 때 이 1이 C로 들어가 「JC」에서 다른 작업의 부분」으로 점프하고 있다(물론 PA0에 리미트스위치 LS를 연결한다면 RRC를 하나만으로 하면 된다).

끝으로 어떤 리미트스위치 LS가 들어왔을 때 그 리미트스위치 LS의 番號에 따라 그 번호의 타스크로 가는 방법의 실험을 한다. 그것은 **그림24**의 프로그램을 마이크로컴퓨터에 넣고 RUN시키면 어떤 리미트스위치 LS도 넣지 않은 경우에는 2進表示部에는 아무 點燈도 되지 않는다. 여기서 먼저 PA0에 (No.1 LS를 ON으로 하여) 5V를 끌어 놓고 RUN시키면 2進表示部에 1이 點燈되고 정지한다. 다음에 PA0는 중지되고 PA1에 (No.2 LS를 ON으로 하여) 5V를 끌어 놓고 RUN시키면 2進表示部에 2가 點燈되고 정지한다.

따라서 「① 8111에서 8113」과 「②8116에서 8119」까지 등에 자기가 希望하는 동작을 시키는 프로그램을 두면 해당 조작을 실행시킬 수가 있다는 것을 알 수 있다. 이것은 8106이나 810B와 마찬가지의 것을 多數 No.3, No.4 ……를 위해 놓고 No.3을 위한 프로그램, No.4를 위한 프로그램 등으로 보낼수가 있는 것이다.

(3) C 포트의 任意의 1개 端子에만 펄스를 送出하는 프로그램

여기서 1개의 端子에만이라고 한 것은 일반 出力命令에 의해서도 펄스는 送出되는데 이 경우에는 예를 들면 그것을 C포트로 하면 C포트 외의 端子 전부에 영향을 미친다. 그러나 이 방법으로는 가령 PC_7에 펄스를 送出해도 PC_6~PC_0의 7개의 端子에는 아무 영향도 미치지 않으므로 때로는 편리한 방법이다.

펄스란 어떤 端子를 한번 1로 하고 다음에 이것을 0으로 하는 것으로 가령 A-D變換칩 등을 개시시키는 등 많은 경우에 필요하다.

이를 위해서는 **그림25**의 프로그램을 實行시키면 된다. 그 이유는 PPI8255의 일반적인 사용 방법의 하나이며 이 사용방법에는 2종류가 있으며 最上位비트를 1로 한 일반 모드設定과 포트設定을 위한 사용방법 외에 最上位비트를 0으로 한 「비트세트

```
8100  3E 90        MVI  A, 90H
  01  D3  03       OUT  03H
  04  DB  00   AA: IN   00H
  06  FE  01       CPI  01H
  08  CA  14 81    JZ   BB
  0B  FE  02       CPI  02H
  0D  CA  19 81    JZ   CC
  10  C3  04 81    JMP  AA
  13  76           HLT
  14  3E  01   BB: MVI  A, 01H
  16  D3  01       OUT  01H
  18  76           HLT
  19  3E 02    CC: MVI  A, 02H
  1B  D3  01       OUT  01H
  1D  76           HLT
```

〈그림-24〉 多數의 리미트 스위치 LS를 判別하여 그 결과에 따라 각각의 작업을 시키는 테스트프로그램

```
MVI  A, 83H
OUT  03H
MVI  A, 0FH  } : 「1」
OUT  03H     }
MVI  A, 0EH  } : 「0」
OUT  03H     }
```

〈그림-25〉 PC의 任意의 端子 하나에 펄스를 送出하는 프로그램

리세트」를 위한 사용법이 있으며 여기서는 그 두번째의 사용방법이 채용되고 있다.

1行째와 2行째는 모드 0, PB를 入力, PC의 上位 4비트를 出力, 下位 4비트를 入力으로 설정하는 것이다. 3行째와 4行째에서 PC_7에 1을 出力하고 5行째와 6行째에서 PC_7에 0이 出力되므로 PC_7에만 펄스가 1回 送出된다.

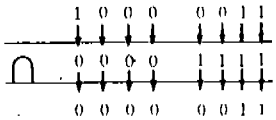
어떤 칩의 始動 등에는 오히려 매우 高速度의 펄스가 좋은데 만일 좀더 늦은 펄스가 필요한 때에는 4行째의 뒤와 6行째의 뒤에 적당한 시간 길이의 웨이트 루틴을 두면 된다.

(4) 마스크를 하는 프로그램

8비트 중의 가령 최상위 비트(MSB라고 한다)만을 0으로 고치려고 할 경우 등이 이 방법의 일례로서 이 경우에는 最上位비트를 마스크한다고 한다.

ANI命令을 실행하면 Acc의 내용과 ANI의 뒤에 있는 數値와의 각각의 비트끼리의 앤드가 취해져 그 결과가 새로 Acc에 들어간다.

예를 들면 **그림26**의 1行째와 같이 Acc에 「1000



〈그림－26〉 마스크하는 일례

0011」가 들어가 있을 때 「0000 1111」와의 앤드를 취했다고 한다. 각 비트마다의 각각의 앤드를 취하여 Acc에 넣으므로 2行째가 0의 비트部分은 모두 0이 되고 1의 비트는 1行째가 그대로 Acc에 들어간다.

결국 2行째가 0의 비트가 마스크된 것이 된다. 最上位만을 마스크하기 위해서는 「ANI 7FH」면 된다.

(5) 어떤 數値의 1/2을 구하는 프로그램

어떤 數値의 1/2을 구하려면 이것을 Acc에 넣고

```
RRC
ANI  7F H
```

〈그림－27〉 數値의 1/2로 한다.

RRC 命令으로 各비트를 1비트씩 우측으로 回轉시키면 된다. 그러나 이것을 실행하면 最下位에서 비트(LSB)가 0이면 문제가 없는데 만일 이것이 1인 경우에는 이 1이 MSB에 들어가 큰 數가 된다. 여기서 이것을 0으로 해야되므로 前述한 방법을사용하여 그림27이면 된다.

(注) (5).00의 部分은 그 경우에 적합한 數値(16進表示)를 넣는다.

(6) 마이크로컴퓨터에서는 「민값」이라고 하는 限界電圧이 있으며 어떤 電圧値 이하는 0으로 판단하고 어떤 電圧値 이상은 1로 판단하도록 되어 있다. 이에 의하여 다소의 電源電圧의 변화나 雜音 등에 의한 誤動作을 방지하고 있다. *

---

* 2分講座 * 〈原子力의 수수께끼〉 ⑪

原子爐에 點火하는 불씨는 무엇인가

原子爐속에 中性子가 있어 核分裂이 일어나면 새로운 中性子가 생겨 連鎖反應으로 계속 運轉을 할 수가 있다 그렇다면 原子爐가 中止되고 있을 때 이를 動作시키기 위한 中性子는 어디에서 얻을 수 있는지. 이 問題를 解決하는 것이 中性子源으로서 爐心의 內部 또는 그 부근에 놓이게 된다.

中性子를 發見하게된 歷史的인 反應으로서 벨리륨에 α 粒子를 접촉시키면 中性子가 放出되는(α, η) 反應이 일어난다. 때문에 例로 라듐, 플로토늄238, 아메리슘241과 같이 α 崩壞를 하여 α 粒子를 放出하는 放射性元素와 벨리륨을 함께 두면 中性子源이 된다 .

큰 原子爐에서는 中性子源의 强度가 1秒間에 數億個의 中性子를 放出하는 程度의 것이 使用된다.

最近에는 特殊한 超우란元素를 工業的으로 入手할 수 있게 되어 칼리폴늄253等이 中性子源으로서 사용된다. 이는(α, η)反應이 아니며 自發核分裂에 의해 放出되는 中性子를 利用하는 것으로서 發電用의 大型爐에 사용되고 있다.

實例로서는 國民학교 아동의 손가락 정도의 크기의 것 2個로 충분하다는 程度로 작은 것으로도 된다. 이 때까지 記述한바와 같이 原子爐의 最初의 運轉에서도 使用할 수 있는 것을 1次中性子源이라고 부르고 있다.

1次中性子源과는 달리 例로 안티몬과 벨리륨을 切半씩 섞은 混合物을 스텐레스鋼等의 管內에 密封하여 原子爐속에 넣어 두면 처음에는 中性子를 내지 않으나, 어느時間 以上 原子爐를 운전한 後에는 안티몬이 放射能을 갖게되어 벨리륨과의 사이에 (α, η)反應이 일어나 中性子源으로서 動作하게 된다.

이같은 種類의 中性子源을 2次中性子源이라고 말한다. 이것을 使用하게 되면 中性子源이 弱하게될 念慮는 없다.

이와같은 現象은 一般家庭에서 잘 있는 瞬間주전자나 가스목욕탕의 點火와 닮았다.

最初의 點火에는 성냥과 電池式 라이타等으로 點火하나 이것이 1次中性子源에 相當한다. 全面的으로 버너에 불이 붙으면 바이퍼스한 細管의 끝에도 불이 붙어, 그後는 버너를 꺼도 細管의 先端의 불은 꺼지지 않으며, 再次 가스의 主流가 흘러 내리면 이 불이 버너에 點火된다.

이 작은 도화선에 相當하는 것이 2次中性子源이다.

原子爐를 中性子源의 中性子에 의해 起動하면 核分裂이 일어나 현격한 差異로 大量의 中性子의 흐름인 中性子사이클이 돌기시작하여 原子爐는 長時間 운전을 계속할 수가 있다.

原子爐의 밖에는 中性子는 거의 없으며 萬一 있다고 해도 短時間에 陽子로 變하기 때문에 外界로 부터 뜻하지 않는 中性子가 原子爐內에 들어와 原子爐가 動作할 念慮는 全然 없다.