

# シーケンス制御パソコンプログラムの効果的作成法と確認法

青森職業能力開発短期大学校

佐々木 進

An Effective Method in Produce and Check of Personal Computer Programs for Sequence Control

Susumu SASAKI

**要約** 機械装置をシーケンス制御する電気的方法は、有接点リレー法、無接点リレー法、プログラムブルコントローラ法及びコンピュータ法等がある。本報告はこの中のコンピュータ法の制御プログラムの作成に関するもので、シーケンス制御パソコンプログラムを効果的に作成する方法と、作成した制御プログラムを確認する制御実験法について具体的に述べている。

有接点リレー法では歴史的に、多くのシーケンス動作の制御回路が考案、実用化されている。この有接点リレーシーケンス制御回路図を参照し、回路開発の省力化を行い、簡単に目的の制御プログラムを作成できるようにした。すなわち、この研究では有接点の制御回路図を出発点とし、制御プログラムの効率よい作成を行っている。

ここでは有接点の制御回路図を基に、三つの方式（流れ図方式、論理演算方式、ブール代数方式と呼ぶ）で、制御プログラム（N88-BASIC）を作成している。また、作成した制御プログラムの有効性を確認するには、自作の入出力実験装置を用いている。その装置についても報告する。最後に、三相誘導電動機の正転・逆転二ヶ所運転制御の例で、制御プログラムの作成から実用機器への適用までについて述べている。

## I はじめに

本報告はシーケンス制御パソコンプログラム（以下、「制御プログラム」という）の作成法とこの制御プログラムの動作の確認法について記述している。内容は制御プログラムの作成法、制御シミュレーション及び制御プログラムの作成法三方式の比較検討となっている。

制御プログラムの作成法では、制御機器となるパソコンに適合する制御プログラムを、三通りのアプローチで考えている。工作機械等をシーケンス制御する制御目的で、古くから使われている有接点リレーシーケンス制御回路図、または新規に作成したものがあるとする。この回路図から、三方式によりそれぞれ単独の考え方で同一の制御動作をする異なった制御プログラムが作成できる。この三方式はフローチャートを利用する方法（以下「流れ図方式」という）、ビットの論理演算を行い演算結果の真偽を扱う方法（以下「論理演算方式」という）及びこの論理演算方式の変形であ

る加積算方法（以下「ブール代数方式」という）である。これらの方式について、それぞれリレー接点を直列、並列、直並列接続に分け詳細に考え方を述べている。また、内部リレー接点及びタイマ接点の扱いについても触れている。

制御シミュレーションでは、制御プログラム開発用に製作した信号入力実験装置及び信号出力実験装置について詳細に記述している。作成した制御プログラムが目的の制御動作をするか確認する必要がある。また、実用機器への適用について、三相誘導電動機の正転・逆転二ヶ所運転制御を例に実負荷への応用を記述している。

最後に、制御プログラムの作成法三方式の比較検討をしている。この結果、いずれの方式でも大差なく制御プログラムの作成、使用ができることが分かった。

## II 制御プログラムの作成法

ここで述べている内容は、機械をシーケンス制御す

るパソコンプログラムを系統だて、効率的に作成する方法である。この制御プログラム作成には、基本として、前述の流れ図方式、論理演算方式及びブール代数方式が考えられる。この三方式別で、リレー接点を並列、直列及び直並列接続回路の要素に分けそれぞれの思考法を述べ、N88-BASICプログラムの例で示している。

また、接点はパソコンI/Oインターフェースボードの入力ポートに接続するような外部接点(スイッチ等)であれば、静止時の接点开(a接点)、閉(b接点)に関係なく、同じ扱い方でプログラムできる。これは、制御の開始時点でそれぞれの接点开、閉状態の初期値で接続されるから、特にプログラム上、配慮する必要がない。しかし、プログラム内部で発生させるリレー接点等については、接点の開、閉の扱いに特別な考え方で対処しなければならない。これらについても述べている。また、タイマ回路の扱いについても述べている。

### 1 流れ図方式

この流れ図方式はプログラムの処理の流れを表す判断記号(◇)を主に用いて、これから制御プログラムを作成する方法である。判断記号での真偽の分岐は記号の下方向が真("y")、左右方向を偽("n")に統一使用する。この流れ図を使用する制御プログラムの作成法について、つぎの(1)～(4)に記述する。

#### (1) 接点並列接続回路

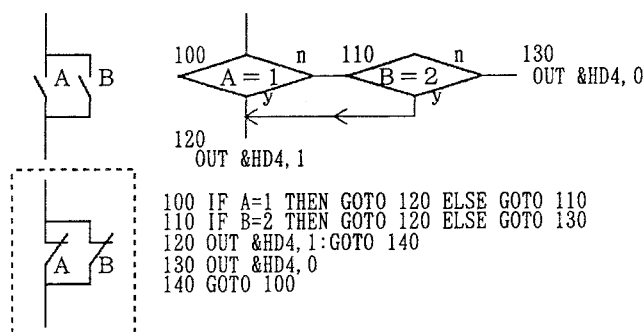


図1 接点並列接続回路(流れ図方式)

外部接点A、Bが並列に接続された回路のときは図1接点並列接続回路(流れ図方式)に示すように、流れ図の判断記号を一接点につき一個の使用で、左右に並列に置いた表現にする。図1は接点が二個の場合である

が、数個の並列接点では数個分の判断記号を並列に並べる。

外部接点AはパソコンI/Oインターフェースボードの入力ポート0ビット目に接続、外部接点Bは1ビット目に接続されてあるとすれば、入力条件A=1(2進法:00000001)、B=2(2進法:00000010)で、それぞれ、入力条件が真の“y”方向分岐で閉接点の表現になる。また、入力条件が偽の“n”方向分岐はそれぞれの開接点情報の表現となる。図1の判断記号でA=1が“y”のときは出力ポート(ここでは&HD4)に1のビット信号を送るが、“n”のときは外部接点Bの入力条件を判断する照合にまわる。外部接点Bのこの照合もB=2の“y”では外部接点Aのときと同様、出力ポートに1のビット信号を送るが、“n”では出力ポートに0(2進法:00000000)のビット信号を送るようにする。この流れ図の使用により制御プログラムを記述すると図1中のBASICプログラム例の100~140行のようになる。前述のように開接点と閉接点とを区別しないでプログラムの作成ができる。これは図1の点線内に示す閉接点接続でも上述したのと同じ流れ図の使用となり、当然、開接点接続と同じプログラムになることを意味する。点線で囲んだ回路については以下同様とする。

#### (2) 接点直列接続回路

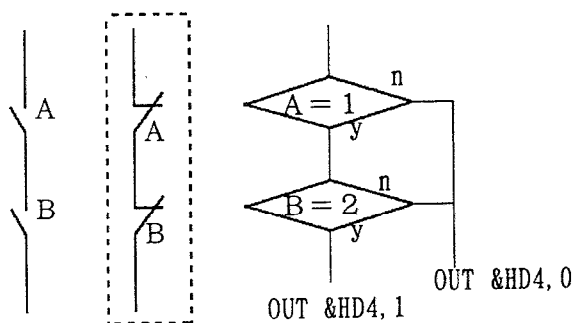


図2 接点直列接続回路(流れ図方式)

また、外部接点A、Bが直列接続回路のときは、図2接点直列接続回路(流れ図方式)に示すように、流れ図の判断記号を上下に直列に置いた表現にする。また、直列接続の接点が数個ある場合は判断記号を数個分、直列に並べた表現になる。これで、上記の(1)接点並列接続回路と同様に制御プログラムは書ける。

(3) 接点直並列接続回路

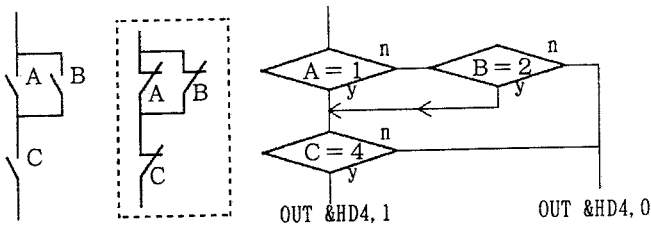


図3 接点直並列接続回路 (流れ図方式)

外部接点A、B、Cが図3のように直並列接続回路のときは上記、図1と図2に示した並列接続と直列接続回路の表現に従い、図3 接点直並列接続回路 (流れ図方式) に示すように並列接続のもの、直列接続のものとを組合わせた表現にする。ここで、外部接点CはパソコンI/Oインターフェースボードの入力ポート2ビット目に、接続されてあるとしてC=4 (2進法: 00000100) を入力条件にしてある。このフローチャートを利用して目的の制御プログラムが作成できる。

(4) 内部リレー接点の扱い

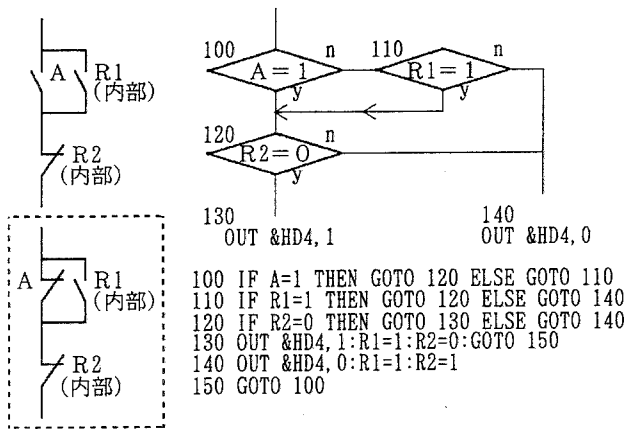


図4 内部接点回路対策 (流れ図方式)

図4の接点R1、R2は内部リレーの接点として制御プログラムで発生させたものとする。これら内部発生リレーの接点を流れ図方式で扱うには特別な配慮が必要であり、これについて述べる。リレー静止時に接点R1は開、R2は閉状態であるものとする。リレー動作時にはそれぞれの接点が反転し、接点R1は閉状態、接点R2は開状態になる。これを利用して、動作時のR1=1、R2=0を内部条件として分岐判断を行う。つまり、内部リレーを扱うには、論理を反転させプログラムする必要がある。このときのフローチャートとプログラム例を図4 内部接点回路対策 (流れ図方式) に示す。

2 論理演算方式

論理演算方式はBASIC命令語の論理演算子OR、AND<sup>(1)</sup>を用いて真偽を判断し、これから制御プログラムを作成する方法である。論理演算方式による制御プログラムの作成法をつぎの(1)～(4)に述べる。

(1) 接点並列接続回路

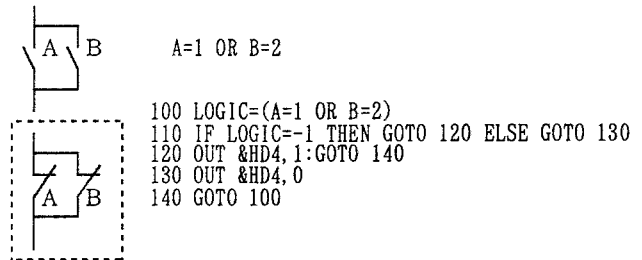


図5 接点並列接続回路 (論理演算方式)

外部接点A、Bが並列接続のときは図5 接点並列接続回路 (論理演算方式) に示すように、論理演算子ORを用いた論理演算式 (A = 1 OR B = 2) の表現にする。この演算子は条件の偽 (0) か真 (-1) を与えると、演算後 0 (偽) か -1 (真) の値を結果として返してくれる<sup>(1)</sup>。そこで、条件A = 1、B = 2を真、A = 0、B = 0を偽にし、二外部接点ABの組合わせの真、偽を調べる論理式は次式 [1] ～ [4] のようになり、それぞれの返す値が分かる。この返す値はビットごとにOR論理演算を行った結果から出たものである。

- (-1 OR -1) = 1 (真)..... [1]
- ( 0 OR 0) = 0 (偽)..... [2]
- (-1 OR 0) = -1 (真)..... [3]
- ( 0 OR -1) = -1 (真)..... [4]

このようにして得られたOR論理演算結果の真、偽を判断することによって、並列接続である接点の開閉について条件分岐ができる。この考え方で制御プログラムを書くと図5の例のようになる。

(2) 接点直列接続回路

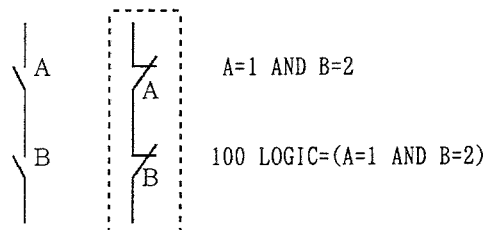


図6 接点直列接続回路 (論理演算方式)

外部接点A、Bが直列接続のときは、図6 接点直列接続回路（論理演算方式）に示すように、論理演算子ANDを用いた論理演算式（ $A = 1 \text{ AND } B = 2$ ）のような表現となる。これも上記の（1）接点並列接続回路と同じ考え方で、外部接点A、Bの組合わせの真、偽を調べる。これは次式 [5] ~ [8] のようになり、論理式はそれぞれのビットごとにAND論理演算を行い、それぞれの結果値を返す。

- ( - 1 AND - 1 ) = - 1 (真) ..... [5]
- ( 0 AND 0 ) = 0 (偽) ..... [6]
- ( - 1 AND 0 ) = 0 (偽) ..... [7]
- ( 0 AND - 1 ) = 0 (偽) ..... [8]

これもAND論理演算結果の真、偽を判断することにより直列接続である外部接点開閉の条件分岐ができる。この考え方で制御プログラムを記述すると図5のBASICプログラム例の100行目が図6で示すように変わる。

(3) 接点直並列接続回路

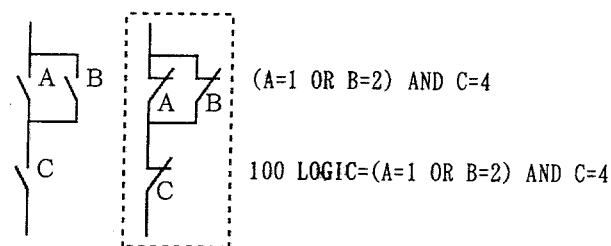


図7 接点直並列接続回路（論理演算方式）

外部接点A、B、Cの直並列接続のばあいの表現は図7 接点直並列接続回路（論理演算方式）に示すような場合、 $(A = 1 \text{ OR } B = 2) \text{ AND } C = 4$ の論理演算式表現となる。これも上記の（1）接点並列、（2）直列接続回路と同じ考え方で、 $A = 1$ 、 $B = 2$ 、 $C = 4$ の真、偽を条件とする論理式の演算結果は次式 [9] ~ [16] のような真、偽の値を返す。

- ( - 1 OR - 1 ) AND - 1 = - 1 (真) ... [9]
- ( - 1 OR - 1 ) AND 0 = 0 (偽) .. [10]
- ( - 1 OR 0 ) AND - 1 = - 1 (真) .. [11]
- ( - 1 OR 0 ) AND 0 = 0 (偽) .. [12]
- ( 0 OR - 1 ) AND - 1 = - 1 (真) .. [13]
- ( 0 OR - 1 ) AND 0 = 0 (偽) .. [14]
- ( 0 OR 0 ) AND - 1 = 0 (偽) .. [15]
- ( 0 OR 0 ) AND 0 = 0 (偽) .. [16]

これらの論理演算結果の真、偽を判断することにより直並列接続である接点開閉の条件分岐ができる。

(4) 内部リレー接点の扱い

図8の接点R1、R2は内部リレーの接点であり、これも前述の1流れ図方式と同様、 $R1 = 1$ 、 $R2 = 0$ と論理を反転させプログラムする必要がある。図8のプログラム例によりこの制御プログラムを示す。

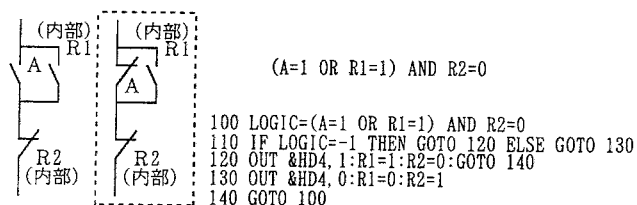


図8 内部接点回路対策（論理演算方式）

3 ブール代数方式

ブール代数方式は論理和、論理積に記号（+）、（\*）を用いて加積算を行い、 $> 0$ かどうかから真偽を判断し、これから制御プログラムを作成する方法である。このブール代数方式による制御プログラムの作成法はつぎの（1）~（4）のようになる。

(1) 接点並列接続回路

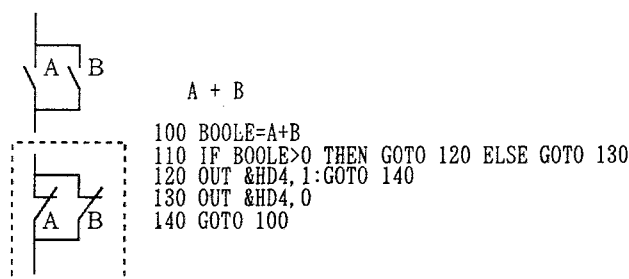


図9 接点並列接続回路（ブール代数方式）

外部接点A、Bが並列接続のときは図9 接点並列接続回路（ブール代数方式）に示すように、加算記号（+）を用いた計算式（ $A + B$ ）の返す値を扱う表現になる。この式の実際の計算は次式 [17] ~ [20] のように加算を行い、結果 $> 0$ かどうかで真偽を判断する。

- ( 1 + 2 ) = 3 > 0 (真) ..... [17]
- ( 1 + 0 ) = 1 > 0 (真) ..... [18]
- ( 0 + 2 ) = 2 > 0 (真) ..... [19]
- ( 0 + 0 ) = 0 (偽) ..... [20]

この計算値が $> 0$ のときを真とし、I/Oポートに出力1を、偽のときは出力0を送るように条件分岐させる。この制御プログラム例を図9に示す。

(2) 接点直列接続回路

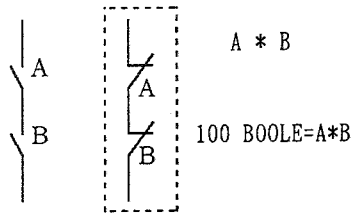


図10 接点直列接続回路（ブール代数方式）

外部接点A、Bが直列接続のときは、図10接点直列接続回路（ブール代数方式）に示すように、積算記号（\*）を用いた計算式（ $A * B$ ）の表現となる。この式の実際の計算は次式 [21] ~ [24] に示すように積算を行う。

$(1 * 2) = 2 > 0$  (真) ..... [21]

$(1 * 0) = 0$  (偽) ..... [22]

$(0 * 2) = 0$  (偽) ..... [23]

$(0 * 0) = 0$  (偽) ..... [24]

この真偽の判定は前述の (1) 接点並列接続回路と同様で  $> 0$  かどうかで行う。この真偽の結果を利用して条件分岐させる。

外部接点A、B、Cの直並列接続の表現は図11接点直並列接続回路（ブール代数方式）に示すように  $(A + B) * C$  の加積計算式となる。接点開閉組合せの実際の計算は次式 [25] ~ [32] のようになる。

$(1 + 2) * 4 = 12 > 0$  (真) ..... [25]

$(1 + 2) * 0 = 0$  (偽) ..... [26]

$(1 + 0) * 4 = 4 > 0$  (真) ..... [27]

$(1 + 0) * 0 = 0$  (偽) ..... [28]

$(0 + 2) * 4 = 8 > 0$  (真) ..... [29]

$(0 + 2) * 0 = 0$  (偽) ..... [30]

$(0 + 0) * 4 = 0$  (偽) ..... [31]

$(0 + 0) * 0 = 0$  (偽) ..... [32]

このように、加積計算式の結果から前述の並列及び直列接続回路同様の真偽判定で条件分岐できる。

(4) 内部リレー接点の扱い

図12の接点R1、R2はこれも前述の二方式と同様、内部リレーの接点である。ブール代数方式での内部リレー接点の扱いは、静止時開接点のR1はそのまま式を作り、閉接点のR2は  $(1 - R2)$  と論理を反転させた値でプログラムする必要がある。つまり、図12の回路の加積計算式は  $(A + R1) * (1 - R2)$  のようになる。

(3) 接点直並列接続回路

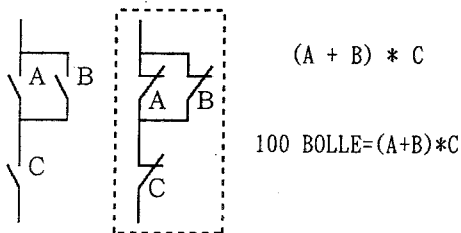


図11 接点直並列接続回路（ブール代数方式）

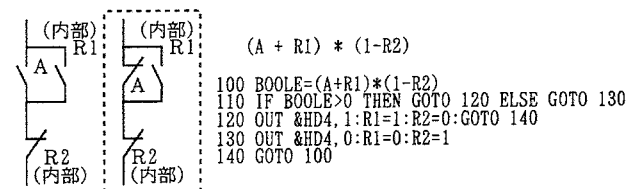


図12 内部接点回路対策（ブール代数方式）

4 タイマ接点の扱い

制御回路でリレー接点の開閉に時限を持たせている

```

100 *SET. N
102
104 '9 ビョウ カノ N スカウト
105 .....
110 TIMES$="00:00:00"
120 T1$=TIMES$
130 SS1=VAL(RIGHT$(T1$,1))
140 FOR I=0 TO 100000
150 T2$=TIMES$
160 SS2=VAL(RIGHT$(T2$,1))
170 IF SS2-SS1<9 THEN 180 ELSE 190
180 NEXT I
190 N=I: RETURN
    
```

(1)プログラム(Nカウント)

```

200 *GET TIME
202 INPUT "DELAY TIME =":DTIME
204 NDT=N*DTIME/9: 'N 〆 *SET. N ヱリ
205 .....
210 TIMES$="00:00:00"
220 T1$=TIMES$
230 SS1=VAL(RIGHT$(T1$,1))
240 FOR I=0 TO 100000
250 T2$=TIMES$
260 SS2=VAL(RIGHT$(T2$,1))
270 IF I<NDT THEN 280 ELSE 290
280 NEXT I
290 RETURN
    
```

(2)プログラム(時限設定)

図13 0.1秒単位の時限設定プログラム例

場合がある。この時限動作をさせる制御プログラムは、パソコンに内蔵のクロックを利用すると時間が正確である。すなわち、システム変数の TIME \$<sup>(1)</sup> を使ったサブルーチンプログラム<sup>(2)</sup> を作成して使用する。

また、この TIME \$ が扱う時間の最小単位は 1 秒である。BASIC コマンドだけによる 0.1 秒単位の時間設定は、図 13 のプログラム例に示すようにする。図 13 の (1) プログラムは FOR I=0 TO N: NEXT I 文の 9 秒間に対する N の値を求めている。この N の値は、使用パソコンの処理速度が影響し機種により異なる。図 13 の (2) プログラムは、N を設定した時とほぼ同じ条件のプログラムにより、N の値から時間解読を行っている。これで、202 行で外部から入力した値 (DTIME) の 0.1 秒単位の時限設定が可能である。図 13 のプログラム例の時間解読誤差は使用パソコンの機種により異なるが、数% (5% : NEC PC-9801 DX) である。これは図 13 の 170 行と 270 行とが異なるのに起因する。従って、図 13 のプログラム例で誤差を含むが、0.1 秒単位の時限設定ができる。

### III 制御シミュレーション

つぎに、制御プログラムの動作を確認するのに必要な資材について述べる。作成した制御プログラムの確認には入力機器 (スイッチ、センサ等)、制御機器 (パーソナルコンピュータ) 及び出力機器 (電動機、ランプ等) に相当する資材が必要である。図 14 制御プログラム開発用資材を示す。

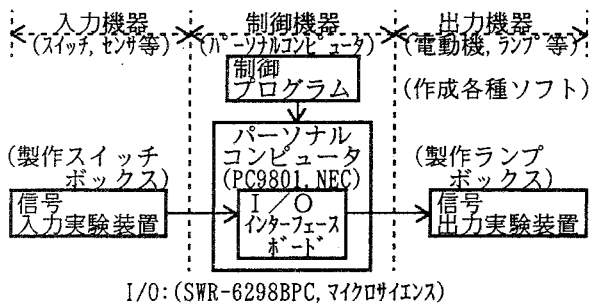


図 14 制御プログラム開発用資材

制御機器として働くパーソナルコンピュータは汎用のもの (PC-9801、NEC) を使用し、これに市販の I/O インターフェイスボード (接点入出力形 (入力 8 接点、出力 4 接点)、SWR-6298BPC、マイクロサ

イエンス)<sup>(3)</sup> を装着してある。入力機器及び出力機器に相当する実験装置は独自のものを製作して用いた。これらの他に目的の制御動作をさせるように、前述の三方式で作成したいずれかの制御プログラムも必要である。

#### 1 信号入力・出力用実験装置の作成

制御実験を効率よく行うには、適切な制御プログラムと取扱い易い実験装置が必要である。制御プログラムは前述のようにして作成できるが、実験装置はパソコン以外に外付するものを準備しなければならない。これは制御プログラムに外部から影響を与える信号入力実験装置及び制御結果を確認する信号出力実験装置である。この目的で作成したものを図 15 信号入力、出力実験装置の外景写真により示す。

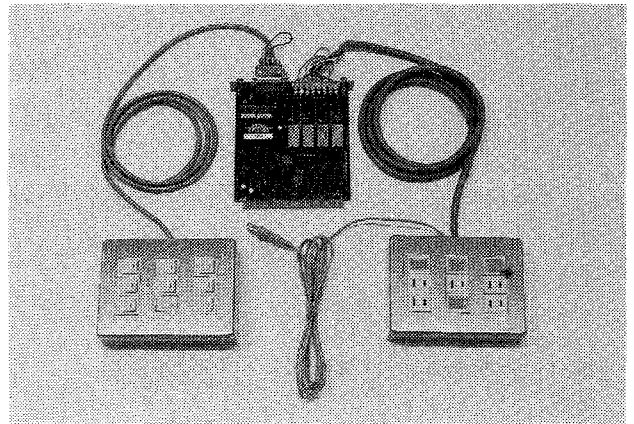


図 15 信号入力、出力実験装置の外景写真

信号入力実験装置はボックス、スイッチ、コネクタ及び信号ケーブル等からの構成である。これらの配置及び内部配線を図 16 信号入力実験装置により示す。

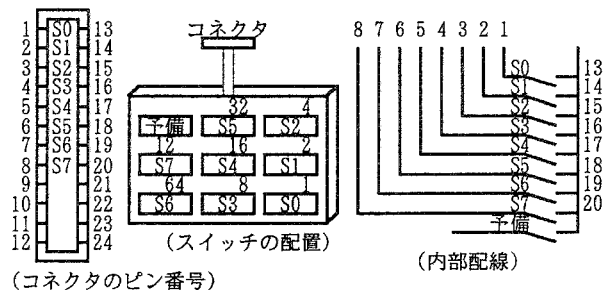


図 16 信号入力実験装置

スイッチ S0 - S7 はそれぞれパソコン I/O ボードの入力側 0 - 7 ビット目に接続できるように、コネクタ内の結線を図 16 の内部配線に示すようにしてある。

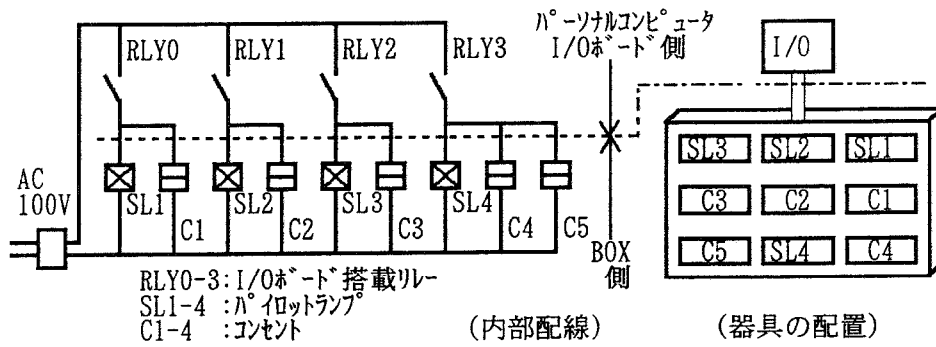


図 17 信号出力実験装置

また、信号出力実験装置はボックス、ランプ、コンセント及びケーブル等から構成されている。これらの配置及び内部配線を図 17 信号出力実験装置に示す。ランプ及びコンセントはパソコン I/O ボードに搭載の继电器 (RLY0 - RLY3) に接続している。この继电器はボードの出力側 0 - 3 ビット目と関連がある。

## 2 実用機器への適用

前述の制御プログラム作成法で制御プログラムを作成し、信号入力、出力実験装置を使用して制御動作を確認できた後、この制御プログラムの最終確認は実際の使用電圧で機器を制御して行えばよい。

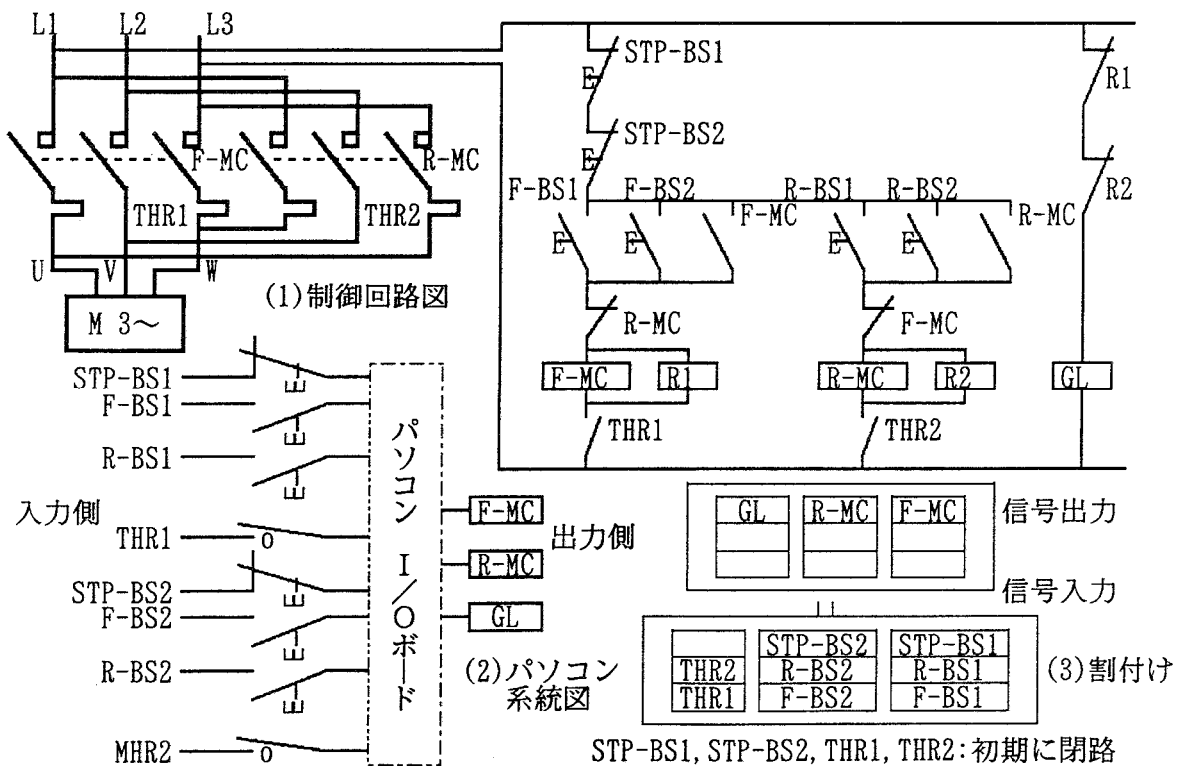


図 18 実用機器適用制御回路 (三相誘導電動機の正転・逆転二ヶ所運転制御)

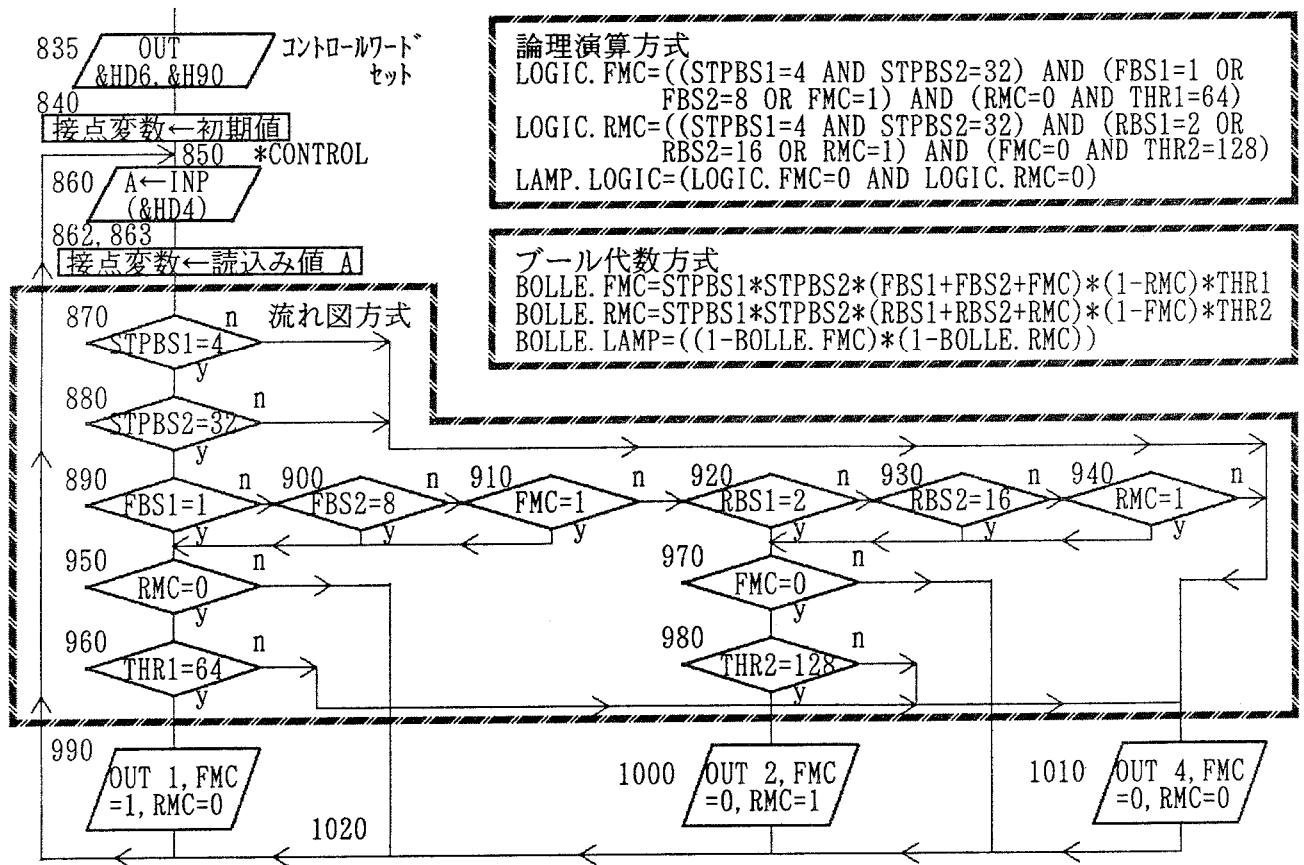


図19 実用機器適用プログラムフローチャート、演算式

```

840 FBS1=0:FBS2=0:STPBS1=4:RBS1=0
842 RBS2=0:STPBS2=32:THR1=64
844 THR2=128:RMC=0:FMC=0
850 *CONTROL
860 A=INP(&HD4)
862 FBS1=A AND 1:RBS1=A AND 2
863 STPBS1=A AND 4:THR1=A AND 64
864 FBS2=A AND 8:RBS2=A AND 16
865 STPBS2=A AND 32:THR2=A AND 128
869 流れ図方式
870 IF STPBS1=4 THEN 880 ELSE 1010
880 IF STPBS2=32 THEN 890 ELSE 1010
890 IF FBS1=1 THEN 950 ELSE 900
900 IF FBS2=8 THEN 950 ELSE 910
910 IF FMC=1 THEN 950 ELSE 920
920 IF RBS1=2 THEN 970 ELSE 930
930 IF RBS2=16 THEN 970 ELSE 940
940 IF RMC=1 THEN 970 ELSE 1010
950 IF RMC=0 THEN 960 ELSE 1020
960 IF THR1=64 THEN 990 ELSE 1010
970 IF FMC=0 THEN 980 ELSE 1020
980 IF THR2=128 THEN 1000 ELSE 1010
989
990 OUT &HD4, 1:FMC=1:RMC=0:GOTO 1020
1000 OUT &HD4, 2:FMC=0:RMC=1:GOTO 1020
1010 OUT &HD4, 4:FMC=0:RMC=0
1020 GOTO *CONTROL
    
```

```

869 論理演算方式
870 LOGIC.FMC=((STPBS1=4 AND STPBS2=32)
AND (FBS1=1 OR FBS2=8 OR FMC=1)
AND (RMC=0 AND THR1=64))
880 LOGIC.RMC=((STPBS1=4 AND STPBS2=32)
AND (RBS1=2 OR RBS2=16 OR RMC=1)
AND (FMC=0 AND THR2=128))
890 LAMP.LOGIC=(LOGIC.FMC=0 AND LOGIC.RMC=0)
900 IF LOGIC.FMC=-1 THEN 990 ELSE 910
910 IF LOGIC.RMC=-1 THEN 1000 ELSE 920
920 IF LAMP.LOGIC=-1 THEN 1010 ELSE 1020
989
    
```

```

869 ブール代数方式
870 BOLLE.FMC=STPBS1*STPBS2*(FBS1+FBS2+FMC)
*(1-RMC)*THR1
880 BOLLE.RMC=STPBS1*STPBS2*(RBS1+RBS2+RMC)
*(1-FMC)*THR2
890 BOLLE.LAMP=((1-BOLLE.FMC)*(1-BOLLE.RMC))
900 IF BOLLE.FMC > 0 THEN 990 ELSE 910
910 IF BOLLE.RMC > 0 THEN 1000 ELSE 920
920 IF BOLLE.LAMP > 0 THEN 1010 ELSE 1020
989
    
```

図20 実用機器適用パソコンプログラム (三方式)



三相誘導電動機を二ヶ所の個別の場所から正転・逆転運転制御を行う例について、この制御プログラムの作成から最終確認までを述べる。これには図18 (1) 制御回路図のような有接点リレーシーケンス制御回路図を出発点とする。最初に図18 (1) 制御回路図から、図18 (2) パソコン系統図に示すように、入力と出力について整理する。つぎに制御プログラムの確認用である信号入力、出力実験装置への信号の割付けを、図18 (3) 信号入力、出力の割付けで示すように行う。

つぎに、制御プログラムの記述について述べる。制御プログラムの作成法で記述した三方式によるプログラムフローチャート及び演算式を図19 実用機器適用プログラムフローチャート、演算式に示す。この図19を利用し図20 実用機器適用パソコンプログラム (三方式) に示すような制御プログラムが書ける。この制御プログラムをパソコンと信号入力、出力実験装置とで、目的の制御動作を確認し、問題がなければ図21 実用機器適用の外景写真に示すような、実際の制御機器をパソコンに接続して制御プログラムを走らせる。この図21は図15、16、17の信号入力、出力実験装置に相当する部分で、この実際の接続関係は図22 実用機器適用実体配線系統図に示す。尚、図21では三相誘導電動機は省略し、接続器への接続で示している。この実用機器で

の実験で目的の制御動作が得られれば、制御プログラムの作成、制御動作の最終確認は終わったことになり、目的とする制御プログラムは完成する。

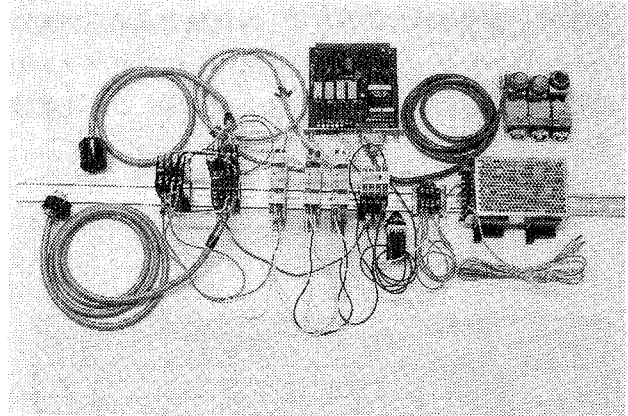


図21 実用機器適用の外景写真

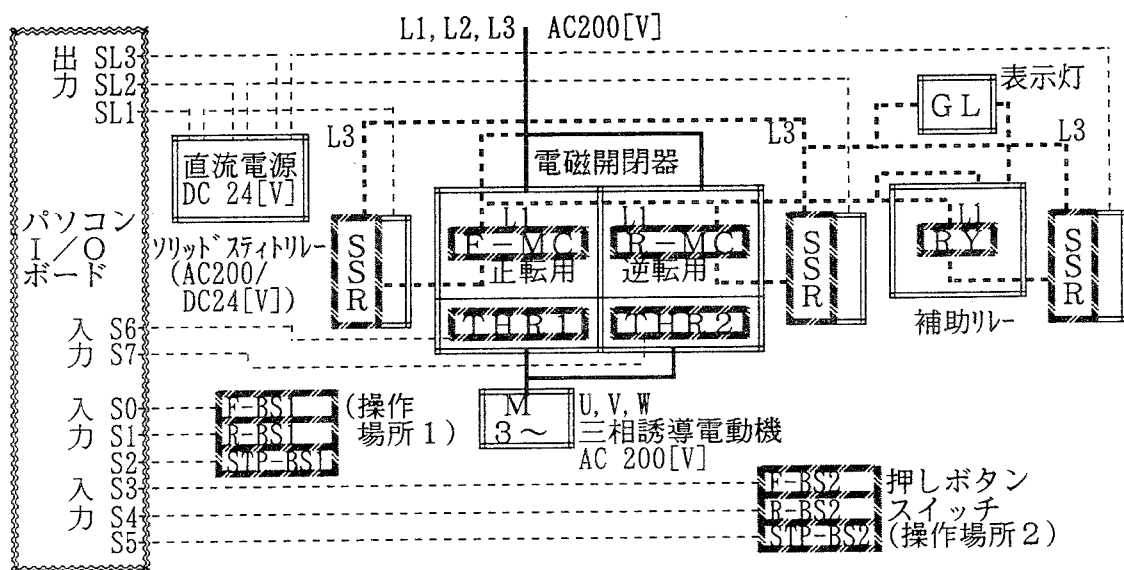


図22 実用機器適用実体配線系統図

## IV 制御プログラムの作成法三方式の比較検討

これまでに、制御プログラムの作成法として、流れ図方式、論理演算方式及びブール代数方式の三方式について述べてきた。この三方式についてプログラムの作成、使用の面から検討を加え比較する。この比較検討した結果を表1 三方式の比較検討に示す。

表1 三方式の比較検討

作成 使用	作成方式	制御プログラムの作成方式		
		流れ図方式 (◇)	論理演算方式 (AND, OR)	ブール代数方式 (+, *)
プログラム	加-図、数式	△	○	○
	書き易い	△	○	○
	短かさ	△	○	○
制御	見通し	○	△	△
	保守	○	△	△
	速応性	○	○	○
教育	動作変更	○	○	○
	精度	○	○	○
	分かり易い	△	○	○
その他	数学的	△	○	○
	内部リレー	○	△	△
	タイマ	○	△	△
	単独作成	○	△	△

◎: 非常によい, ○: よい, △: 普通

比較検討の項目はプログラム、制御、教育及びその他としている。この項目別に各方式の特長を順に述べる。項目プログラムのフロー図、数式についてはブール代数方式が他の方式より分かり易く、そのため短いプログラムの記述で終わる。だが、プログラムの見通し、保守の面は流れ図方式が他の方式より優れている。項目制御について、制御の速応性、動作変更及び確実に動作する精度に三方式とも大差がない。これはプログラムがしっかりしていればいずれの方式でも大差がないことを示している。教育面について、流れ図方式は図形を使用する関係から分かり易い。だが、ブール代数方式の方が数学的である。項目その他として、論理演算方式及びブール代数方式はプログラムを作成するのに、論理演算部分以外はフローチャートを必要とし、それによりプログラムの作成が効果的に行える。すなわち、流れ図方式はこの面から図だけでプログラムの単独作成ができる。

制御プログラムの作成について、三方式はそれぞれ長短があるが、いずれの方式でも大差なく、制御プログラムの作成、使用ができる。これは制御プログラムを知るための教材として評価できる。

## V おわりに

制御プログラムを流れ図方式、論理演算方式及びブール代数方式の各方式で作成できるのが明らかになった。これで、基本的にはすべての有接点リレーシーケンス回路が制御プログラム化できる。また、この三方式のいずれでも大差なく、制御プログラムの作成、使用ができることが分かった。作成した制御プログラムの中間確認に信号入力、出力実験装置を使えると便利であり、この実験装置の作成、使用について具体的に述べてきた。さらに、具体的な制御対象を設定した、実用機器を使用する場合の制御プログラムの作成、実験について述べてきた。このように、制御プログラムの作成法に関して記述された文献は余り多くない。これらの内容は制御プログラムの作成、使用を技術的に解説し、実験する専門教育及び能力開発セミナー等の利用が期待できる。

制御プログラムはこのように作成、確認実験できるが、プログラムには気付かないバグがあるかも知れない。この防止策として三方式を併用し、三方式の出力データが合致したときのみ出力するという方法も考えられる。この三方式併用の効果がどのようなものであるか、今後種々検討して行きたい。

## 参考文献

- (1) 日本電気 (KK), BASIC リファレンスマニュアル PC-9801VX, P12-14, 1981
- (2) 太平洋工業 (KK), 機械に知力をつける パソコンによる制御実習, 日刊工業新聞社 (KK), P86-87, 1985
- (3) マイクロサイエンス (KK), SWR-6298BPC 取扱説明書, P1-2, 1986