

《マイ・リレーのしおり》

プリンタ端子よりの信号でリレースイッチ^(注1)をON・OFFします。

MSXの場合、BASICのプログラムで次のようなコマンドでリレーを動かします。 (MSXのプリンターのアドレスは、&H91)

```
A=&B01000010
```

(マイ・オートメーション・ツールのためのプログラム 参照)

```
OUT &H91,A
```

Aのbitパターン^(注2)により、8個のリレーのON・OFFが決まります。

例えば、 A=&B0 1 0 0 0 0 1 0

リレーのチャンネル ch7 ch6 ch5 ch4 ch3 ch2 ch1 ch0

スイッチ 切 入 切 切 切 切 入 切

ch0を入、ch1を切、ch2を切、ch3を入のように決まったスイッチの入・切のパターンで使うのでしたら、今の例のように直接2進法で変数に代入してやればよいのです。

初めに ① 入 切 入 切 入 切 入 切

次に ② 切 切 切 切 入 入 入 入 のようにスイッチを変えたい時は

```
A=&B10101010
```

```
OUT &H91,A
```

```
FOR i=1 TO 1000:NEXT(注3)
```

```
A=&B00001111
```

```
OUT &H91,A
```

このような命令で①の状態から、②の状態に変わります。まずはこれで充分でしょう。しかし、この方法ではいつも8個のスイッチの入・切のパターンが決まっている時は構わないのですが、8個のスイッチの入・切の組合わせは、全部“切”から全部“入”まで256通りの組合わせがあるのです。8個のリレーをよりうまく使いこなしたい人には、少しわずらわしいですね。

他のスイッチの状態を変えないで、ch2のスイッチだけ変えたいという人のために、こんな演算は御存知ですか？

```

      1 0 1 0 1 0 1 0
AND) 0 0 0 0 1 1 1 1
-----
      0 0 0 0 1 0 1 0
( 5 4 )

```

```

      1 0 1 0 1 0 1 0
OR)  0 0 0 0 1 1 1 1
-----
      1 0 1 0 1 1 1 1

```

プログラム

```

A = &B10101010
N1 = &B00000100
A = (A OR N1)
OUT &H91, A
FOR i=1 TO 100:NEXT
F1 = &B11111011
A = (A AND F1)
OUT &H91, A

```

説明

A = 10101010 だった時、
 OR) 00000100
 A = 10101110 になる。
 ch2のスイッチON、他は変化ない。
 時間かせぎ
 A = 10101110
 AND) 11111011
 A = 10101010
 これで元にもどる。

つまり、
 N0 = &B00000001 : F0 = &B11111110
 N1 = &B00000010 : F1 = &B11111101
 N2 = &B00000100 : F2 = &B11111011
 N3 = &B00001000 : F3 = &B11110111
 N4 = &B00010000 : F4 = &B11101111
 N5 = &B00100000 : F5 = &B11011111
 N6 = &B01000000 : F6 = &B10111111
 N7 = &B10000000 : F7 = &B01111111 という2進数の入っている変数を作っておいて、

ch3を入の時、 A=(A OR N3):OUT &H91, A

ch2を切の時、 A=(A AND F2):OUT &H91, A のようなプログラムを作ればいいですね。

このように、目的のひとつのbit(ひとつのスイッチ)だけを変更したい時に使うN1やF1のマスクパターンの事をマスクと呼んでいます。知っておくと得するかもしれません。もっと頭を使いたい人は、計算で目的のビットパターンを作ってやればいいのですが、このように目的のビットパターンを用意しておいて、AND・ORを取りながら使うのは、後でプログラムを見た時もわかりやすいので便利です。

《もっと楽をしたい人へ》

モーターの正逆回転などのように、2つ以上のリレーを組合わせて同時に変化させたい時など、その2bit用のマスクパターンを用意しておくと便利です。

M0=&B11111001

M1=&B00000010

マスクパターンをセット

M2=&B00000100

M3=&B00000110

A=A AND M0:A=A OR M1:OUT &H91,A

～他のスイッチがどんな状態でもch2が切、ch1が入になる。

A=A AND M0:A=A OR M2:OUT &H91,A

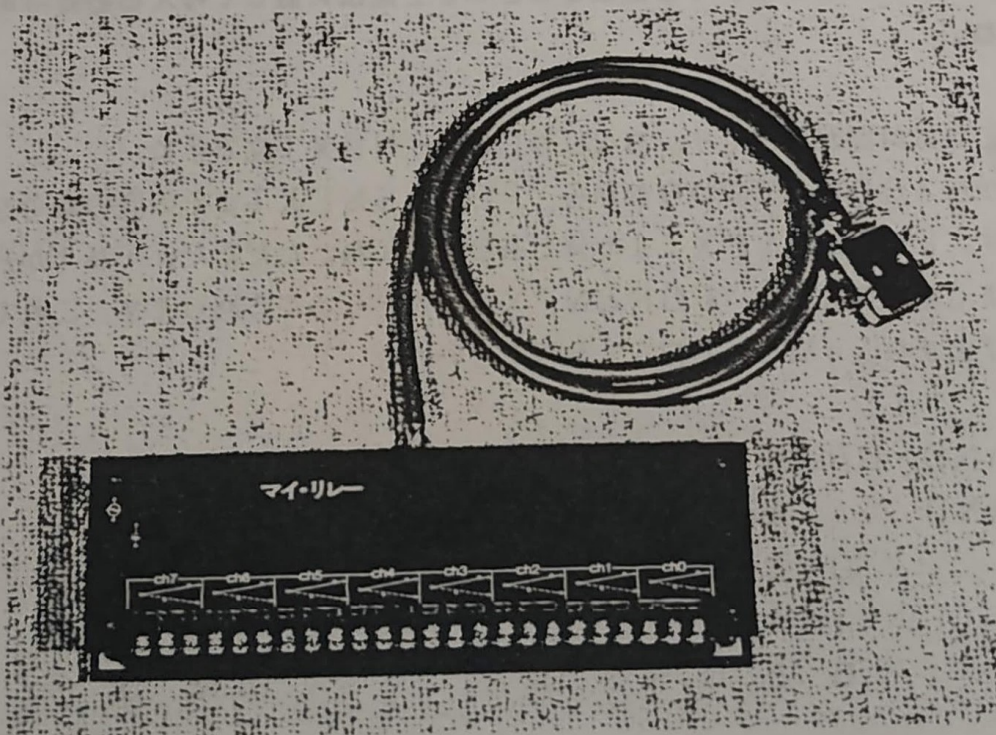
～他のスイッチがどんな状態でもch2が入、ch1が切になる。

A=A AND M0:A=A OR M3:OUT &H91,A

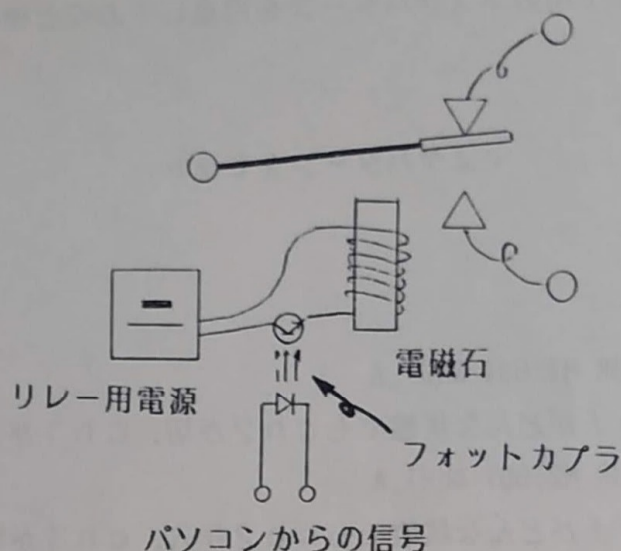
～他のスイッチがどんな状態でもch2が入、ch1が入になる。

A=A AND M0:OUT &H91,A

～他のスイッチがどんな状態でもch2が切、ch1が切になる。



注1》リレースイッチとは電磁石で入れたり切ったりするスイッチのことです。



注2》b i tパターン

私たちが使っている数は、10進法だということは御存知ですね。同じ数を2進法(0と1)で表したものをb i tパターンとってます。b i tとは、0か1の状態しかとりえない最少の数の単位で、2進数の1ケタのことです。BASICでは“&B01000010”というように、表すとb i tパターンがそのまま変数に代入できます。10進法をわざわざ2進法に直して考える必要はありませんね。

注3》これは時間かせぎです。おなじみFOR~NEXT文ですね。

iが1から1000まで変わりながら、FORからNEXTまでの間の命令を実行します。FORからNEXTまで何もありませんね。ただただ、iが1から1000まで変わりながら、FORとNEXTの間をぐるぐる回ります。これで時間が稼げますね。(MSXで約1.5秒ぐらいかな。)

注4》AND・OR論理

AND	1	と	1	->	1	OR	1	と	1	->	1
	0	と	1	->	0		0	と	1	->	1
	1	と	0	->	0		1	と	0	->	1
	0	と	0	->	0		0	と	0	->	0

最後に一つ、

他のいろいろな目的のプログラムとちがって、制御プログラムはタイミングがとてゝ大事になります。例えば、 $2+2$ はどんなに時間をかけて計算しても、たいてい4になります（たまにドジもある）。しかし、スイッチの制御はそうはいきません。

例えば、

```
OUT &H91,1:OUT &H91,0:OUT &H91,1
```

“入”の状態から一瞬，“切”，そしてまたすぐに“入”になるはずです。

しかし、このプログラムでは動作速度がついていけなくて、リレーは動かないと思います。まるでこの命令のかわりに、 $C=3+3$ なんていうようなオンチな命令を書いてしまったのと、みかけは全然変わらない（動かない）のです。

また、

```
OUT &H91,1
FOR i=1 TO 1000:GOSUB 100:NEXT
OUT &H91,0
FOR i=1 TO 1000:GOSUB 100:NEXT
OUT &H91,1
```

のように、時間かせぎのついでに何か処理（GOSUBの中で）をしたとします。

このGOSUB 100の処理が、1回、 $1/100$ 秒ぐらいで済んだとすれば、10秒でスイッチが“入”“切”します。しかし、たまたま作者の意に反して $1/10$ 秒かかってしまったとします。するとこのプログラムは、100秒間だんまりを決めこみます。たいていの人は27秒ぐらい待つと、これはプログラムが暴走してしまったと思うにちがいありません。気の短い人は多分ここで、ストップキーを押して、どこが間違っているのだ、おかしい、とマイ・リレーを蹴とばすでしょう。気の長い人は、すっかりプログラムが動いていることを忘れていて、100秒後にいきなりスイッチがきれて、びっくりしてとびあがるかもしれせん。

人の感覚の幅というものは、自然の時間の流れの中ではきわめてせまいものです。0.01秒での変化も、数百秒での変化もみかけは全く同じ（動かない！壊れたかな？）ということもあり得るのです。

このことは、例えば、1秒だけ通電するなどという制御が必要な時、非常に大切になります。ちょっと間違えてスイッチが入りっぱなし、2～3秒で模型のコイルを焼き切ってしまった、というような事もあり得ます。（この失敗は実在する！）このような場合には、必ず“入”“切”の命令をセットしておくように心がけましょう。

```
A=A OR N2:OUT &H91,A :FOR i=1 TO 1000:NEXT :A=A AND F2:OUT &H91,A
```

また、模型の電源をつなぐ前に、マイ・リレーのランプの点滅で動作を確認しましょう。

時間稼ぎのやり方 （時間の関数などを使いこなせる人はどうぞ勝手に）

```
PRINT TIME
```

～時間の表示

```
FOR j=1 TO 100:FOR i=1 TO 100:NEXT:NEXT
```

```
PRINT TIME
```

～時間の表示

（TIME関数は1/60秒ごとに1ずつ増加します。）

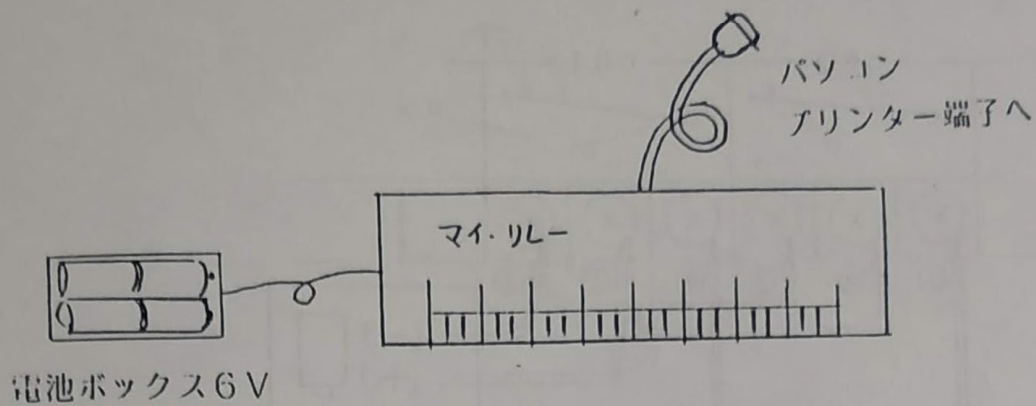
これをrunします。

FOR文の実行にかかった時間を計算して中の100の値を加減して、自分で使う時間かせぎのルーチンをつくっておきます。

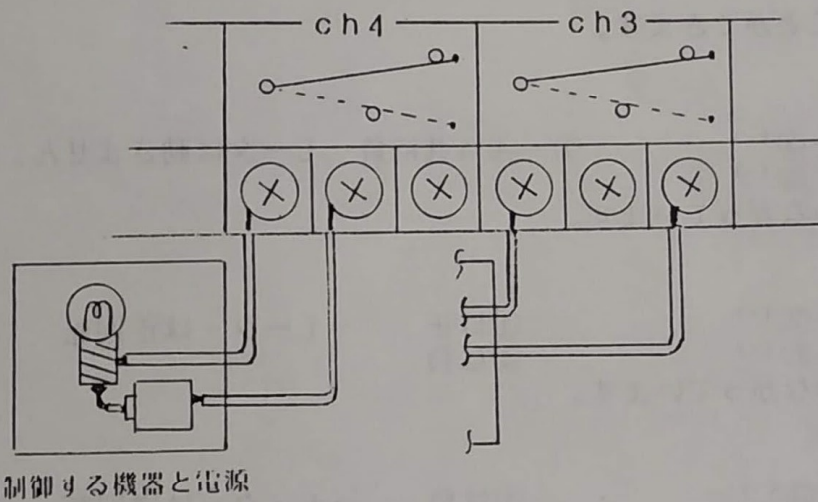
例えば1秒をつくる時は、100秒ぐらい計ってその100分の1をつかいます。これも1ケタでも間違えると、こんな簡単なプログラムでもみている間には終わりません。“暴走だ！”と騒がないで下さい。因みにMSXでは、このプログラムで15秒ぐらいかかります。従って、1秒待つためには、マシンによっても若干異なりますが

```
FOR I=1 TO 620:NEXT という命令をはさめばよいでしょう。）
```


マイ・リレーの接続のしかた



1チャンネルを使用して
電球点灯の制御例

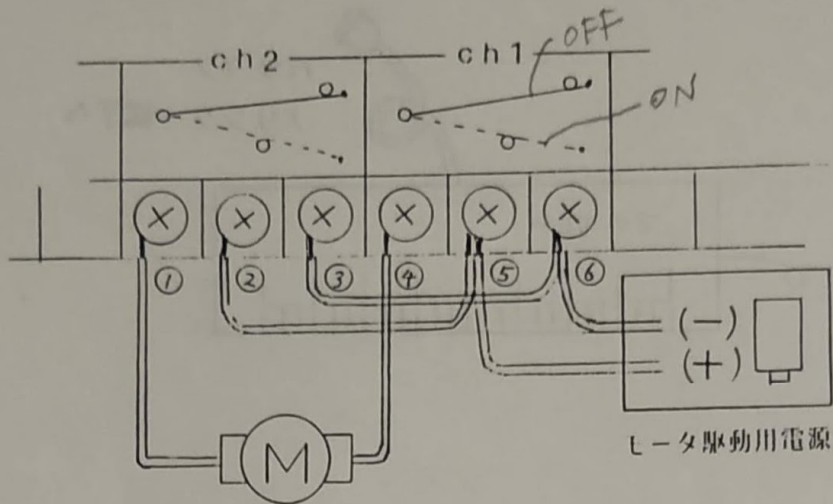


ch4 の例
ONの時ランプ点灯
OFFの時ランプ消灯

ch3 の例
ONの時ランプ消灯
OFFの時ランプ点灯

(注) 使用する電源は、どんなもの(電池、DCアダプター、その他)でもかまいません。
(max:125V/AC, 1A 30V/DC, 2A たいていの模型はこれでOK!
もし、それ以上の大容量のものを動かす時は、当装置で大容量用
リレースイッチを制御して下さい。)

モータ正逆回転の制御例



*モータの正・逆回転制御

逆回転させたりすることができます。

最初

スイッチ

④—⑥(-)

がつなが

がつながっています。

①・④は共に負　モータは動きません。

次に

スイッチ

(4) — (6) (-)

がつなが

がつながっています。

①は正

④は負

→モータは正回転

そして

スイッチ

④—⑤ (+)

がつなが

がつながっています。

①は負

④は正

→ 七 - タ - は逆回転