

# プログラマブルタイマー基板 KP-TMAR08 取扱説明書 応用編(結線マクロ)

第 1 版 190416

## ■はじめに

この説明書は「プログラマブルタイマー基板」(KP-TMAR08)の応用編です。  
本基板には出荷時に予め、汎用的なタイマーとしての機能がプログラムされています。  
出荷時に書き込まれている機能については、基本編の説明書をご覧ください。

本基板のタイマーや他の動作は、このマクロ言語(以下、結線マクロ)による記述で行われています。  
結線マクロを利用することで、本基板の機能を出荷時と異なる機能へと変更することができます。

## ▼プログラマブルタイマー基板 KP-TMAR08 製品ページ

<http://prod.kyohritsu.com/KP-TMAR08.html>



## ▼出荷時の結線マクロ

出荷時の結線マクロは弊社の Web ページからダウンロード可能です。出荷時の機能に戻したい場合は、下記ページにて配布の各結線マクロデータをご利用ください。

またこれらのデータは、本基板で動作させる目的には自由に編集変更してご利用頂けます。

結線マクロダウンロードページ

<http://prod.kyohritsu.com/PROGTIMER/index.html>



## ▼ご注意

結線マクロの動作は十分に検証していますが、全ての機能の組み合わせを試験していません。

自由に組み替え可能な反面予期しない動作が発生する可能性があります。

動作検証を十分に行った上でご利用ください。

なお、共立電子産業は、結線マクロの組み換えによって発生するいかなる損害についても責任を負いかねます。

また、該当機能を実現するプログラムの瑕疵により動作説明と不一致が発生する可能性があります。  
共立電子産業では、結線マクロの作り方のサポートや個別の結線マクロの動作に関するご質問への回答はいたしかねます。ご理解の上、ご利用ください。

## ■結線マクロによる動作

プログラマブルタイマー基板の真髄はこの機能にあります。

基板の基本動作を自由に組み替える事ができます。

### (1) 基本的な考え方

本基板は、入力と出力を持ったCPU基板として作られています。

これらの IO(資源)を組み合わせて動作させたい機能を実現します。

タイマーはこの基板の基本コンセプトですが、タイマーを実現するためには、トリガやリセット用の入力とタイマー動作の出力が必要になります。

これらの機能(I/O機能)を汎用的に分離し、タイマーはソフト的な仮想タイマーとして実現しています。

外部から本基板に対しての入出力(I/O機能、I/O端子)は下記の通りです。

- ・ フォトカプラ入力 4点
- ・ モード設定および論理切り替えに使えるジャンパーピン 6点
  - \*ジャンパーピンによる論理指定以外に、外部から論理信号(5V系 TTL信号)を入力する端子としても使用可能です。
- ・ リレー出力 4点
- ・ アナログ値を発生させる半固定抵抗 4点

これらのI/Oを有意に接続させるのが結線マクロです。

外部からの信号と内部の仮想的なタイマーやその他のソフト的な部品を接続させますので、結線マクロと呼びます。

動作速度の制限を除けば、一般的な部品を組み立ててハンダ付けで結線して欲しい機能を実現するのと同じです。

本基板は、Mode 0～3の4種の動作モードを搭載していますが、結線マクロを作成して書き換えることで、このモード単位で1種のみ動作を変更したり、4種とも自作の動作モードに置き換えて使用したりすることができます。

## (2) 結線マクロ(動作する機能)の書き換え方

結線マクロの書き換えには、PCとの接続が必要です。

KP-TMAR08には、電源入力と結線マクロ転送を兼ねたUSBシリアル端子を装備しています。USB端子をPC(Windows,Mac,Linux等)のUSBポートに接続し、USBシリアルとして認識させる事で、PCと基板間の通信を行います。

機能の設定は結線マクロの形式でテキストデータとして作成し、本基板に転送(送信)します。文法的に正しい動作マクロを受信した本基板は、内蔵の記録エリアに結線情報を保存します。保存された結線情報は電源が切られても保持されます。

再起動(電源を入れ直すかRESETボタンを押す)すれば、記録された新しい結線マクロによる動作を行う事ができます。書き込みが終わればPCは不要です。

### [1. デバイスドライバの組み込み]

本基板はArduino UNO等のマイコンボードと同様、USBシリアルと呼ばれる機構を通してPCと接続します。PCに本基板を接続の間は、シリアルポートと呼ばれる通信用の切り口が作成されます。このポートを用いて、PCから本基板上に通信(結線マクロの転送)を行います。

#### ▼Windowsをお使いの場合

- インターネットに接続されている環境で実行してください。
- 本基板を接続する前に、「デバイス マネージャー」を起動してください。
- デバイス マネージャーの中に「ポート(COMとLPT)」の項目が存在する場合は展開してください。展開したリスト内に「通信ポート(COM1)」など、COM□(□には番号が入ります)が表示されますので、現在の番号をメモしてください。  
「ポート(COMとLPT)」の項目が存在しない場合は次に進んでください。
- 本基板をUSBに接続してください。  
接続後、デバイスの認識中を表す文言が表示される場合もありますが、数分でデバイスドライバの組み込みが完了します。
- デバイス マネージャーの「ポート(COMとLPT)」内に、先程メモしたCOM番号以外に、新たに「USB Serial Port (COM□)」(□には番号が入ります)が増えていれば成功です。  
この□に入る番号が、接続したPCにおける本基板のポート番号となります。
- デバイスマネージャの項目に黄色の[?]と「不明なデバイス」の項目が出てきた場合、デバイスドライバの自動組み込みに失敗しています。この場合は手動でデバイスドライバを組み込む必要があります。下記「USBシリアルコンバータのデバイスドライバ」に移り、手動でインストールしてください。

#### ▼Mac、Linuxをお使いの場合

- 本基板を接続する前に、「ターミナル」を起動してください。  
(Linuxでは「端末」という名前の環境もあります)
- 現れたターミナル画面に、下記のコマンド文字を入力し、[Enter]を入力して実行してください。  
`ls /dev/tty*`
- 画面上に、/dev/tty から始まる結果が出力された場合は、その結果をメモしてください。  
全く表示されず次の入力待ち表示が出た場合は、そのまま次に進んでください。
- 本基板をUSBに接続してください。
- 再度ターミナル画面に、先程と同じコマンドを入力し、[Enter]を入力して実行してください。  
`ls /dev/tty*`
- 先程メモした結果の項目以外に、新たに /dev/tty から始まる項目が増えていれば成功です。  
この項目の名前が接続したPCにおける本基板のポート名となります。
- コマンドを実行しても項目が増えていない場合は、デバイスドライバの自動組み込みに失敗しています。この場合は手動でデバイスドライバを組み込む必要があります。  
4ページ「USBシリアルコンバータのデバイスドライバ」に移り、手動でインストールしてください。

#### ▼USBシリアルコンバータのデバイスドライバ

- 本基板は、USBとシリアルコミュニケーションを仲介するUSBシリアルコンバータチップとして、FTDI社の「FT231XS」を使用しています。
- ドライバの自動組み込みが行われなかった場合は、下記URLより、各環境に合わせたデバイスドライバをダウンロードし、インストールしてください。

<https://www.ftdichip.com/Drivers/VCP.htm>

#### [2. ターミナルソフトの導入]

シリアル通信ではターミナルと呼ばれるソフトウェアを通して文字列を送ったり、表示したりします。PCと本基板とで通信を行い、結線マクロを転送する際には、このターミナルソフトをご用意頂く必要があります。

各PCで使えるターミナルソフトは下記のものが存在します。

- Windows : Tera Term など
- Mac : CoolTerm など
- Linux : GTKterm など

#### [3. 結線マクロの記述]

##### ▼結線マクロの記法

結線マクロの各行は、信号を受け取る機器(入力機器)と送り出す機器(出力機器)の二つで表現されます。

まず第1のパラメータに、信号を受け取る機器の名前を記述します。

次に「,」(半角記号のカンマ)、

その後、第2のパラメータに、信号を送り出す機器の名前、または数値のいずれかを記述します。

行の末尾には、「;」(半角記号のセミコロン)から改行するまでの間に、コメント(自由記述の注釈文)を必要に応じて追加できます。

例えば、フォトカプラ入力の0番から入った信号に従ってリレーの0番をON/OFFさせるだけの結線マクロは

---

```
RY0, PH0 ; フォトカプラ 0 をリレー 0 に接続
```

---

の一行だけです。これだけでもりっぱな機能を果たします。

(フォトカプラによる絶縁入力のリレー装置として機能します)



この記述の解説は下記の通りです。

- ・「RY0」は、信号を受け取る機器としてリレーの0番を指定します。
- ・「,」は、前後の単語(RY0)(PH0)を区切るために入力する文字です。
- ・「PH0」は、信号を送り出す機器としてフォトカメラの0番を指定します。
- ・「;フォトカメラ0をリレー0に接続」の部分は、この行がどのような意味を示しているものであるか理解しやすくするために記載したコメント(注釈文)です。省略していても、他の内容が書かれていても結線マクロの機能にはなんの影響も与えません。
- ・「;」はコメントの開始を宣言する文字です。「;」からその行を改行するまでの部分がコメントとなります。

※RY0やPH0 といった機器名および、使用可能な機器の種類については、付録「結線マクロの動作」にて全て掲載していますので、結線マクロを作成する際に参照してください。

※半角スペースは無視されます。

こういった1行の記述が、入力機器に対して出力機器を接続する1本の結線に相当します。これを機能全体を構成するために必要な数だけ、複数行にわたって作成していきます。

#### ▼ブロックの指定 (BLOCK0,1,2,3,ENDB)

本基板は動作モードをMode 0 ~ 3までの4種から選択できますので、作成した結線マクロを何番のモードに適用するか指定する必要があります。

作成した結線マクロ本体を「BLOCK0」(番号は0, 1, 2, 3のいずれか)~「ENDB」の行で挟む事で、作成した結線マクロを記録するモード番号の指定が行われます。

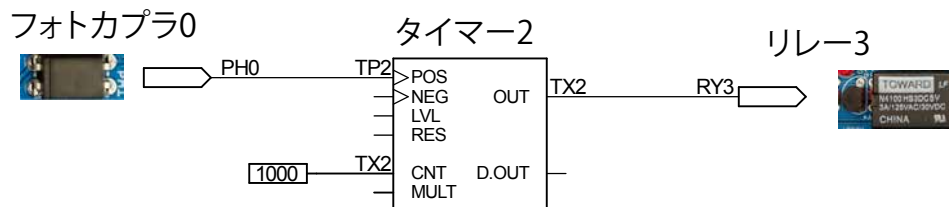
下記の例は、BLOCK0(動作モード0番)に結線マクロを書き込む際の例です。

---

```

BLOCK0           ; 動作モード0のブロックを開始します
  TP2, PH0       ; タイマー2をフォトカメラ0の入力立ち上がりエッジでトリガする
  TX2, 1000      ; タイマー2の設定値を1000(10秒)にセット
  RY3, TY2       ; タイマー2の出力をリレー3に接続
ENDB             ; ブロックを終了します
  
```

---



BLOCK0、ENDBはコマンドの一部であり、結線情報ではありません。

BLOCK0 は、これ以後の結線情報は動作モード0に関するものであることを宣言するための記述です。

以後、動作モード0での結線マクロの中身を記述します。

結線マクロが終わったらENDBで動作モード0に関するブロックはここまでであることを宣言します。

ENDBコマンドを受信したタイミングで、結線マクロの受け付けを終了し、CPU上の保存領域への上書き更新を行います。その際、本基板から PC へのシリアルメッセージに

「Update.....」→「OK.」という進捗表示が行われます。

※更新作業中は、次のコマンドや結線マクロの入力を受け付けません。続けて他のブロックを変更する場合は、「OK.」が表示されるまで待機してから転送してください。

※BLOCK0 ~ 3のコマンドで開始宣言されていない結線マクロは有効になりません。

必ず BLOCK0 ~ 3コマンドを忘れないように入力してください。

#### ▼動作の停止と再開(STOP,RUN)

結線マクロは、タイマー基板の停止中にのみ書き込むことができます。

KP-TMAR08は電源投入またはリセット直後、ジャンパーM0, M1にて指定されたモードでの動作が開始されますので、結線マクロを書き込む前に動作を停止する必要があります。

停止するためには、STOP コマンドを送信します。下記のように、

---

STOP

---

のみ記載された行を送信することで、動作が停止します。

本基板からは「Stop」というメッセージが出力されます。

再開するときは、RUNコマンドを送信します。下記のように、

---

RUN

---

のみ記載された行を送信することで、ジャンパーM0, M1で指定されたモードの動作が開始します。本基板からは「Run <Mode 0>」(モード番号はジャンパー設定に依存します)というメッセージが出力されます。

結線マクロをテキストファイルから一括で転送する際は、ファイルの先頭にSTOP、末尾にRUNコマンドを予め入れておくことで、自動で書き込み前の停止と書き込み後の再開ができます。

---

STOP

BLOCK0 ; 動作モード0のブロックを開始します

TP2, PH0 ; タイマー2をフォトカプラ0の入力立ち上がりエッジでトリガする

TX2, 1000 ; タイマー2の設定値を1000(10秒)に

RY3, TY2 ; タイマー2の出力をリレー3に接続

ENDB ; ブロックを終了します

RUN

---

※動作中には、タイマー処理を優先するため、STOP以外の一切の入力を受け付けません。  
まずSTOPコマンドを送信してください。

## 付録：結線マクロの動作

結線マクロはKP-TMAR08基板が持っている資源を有機的に結合して動作させる、一種のインタープリタプログラムです。

記述された結線情報に従い、入出力やタイマーの動作を制御します。

### ▼結線マクロの詳細と書き込み

本基板は、信号を発生させる部品(出力機器)と受け取る部品(入力機器)を持っています。またタイマーや論理ゲート、選択回路、演算回路などソフトウェアで実現する論理的な部品(内部機器)も用意しています。

これらの機器間の端子を接続する事で有機的な機能を果たす事ができます。

この接続を指示する機能が結線マクロです。

結線マクロは一種の命令を書き込んだ文章で、ターミナルソフトウェアに直接入力するか、テキストファイルとして作成し、それを転送します。

この文章(テキストファイル)を基板にUSBシリアルポートを通して転送する事により、結線の指示を与えます。

### ○シリアルポートの設定

通信速度：9,600bps

データ：8ビット、パリティビット：無し、ストップビット：1ビット

### ○結線マクロの記法

結線マクロ行中に存在する半角スペース、タブ文字は無視されます。

改行はCRコードまたはCR+LF コードです。

構文は下記の通りです。

---

Dest, Src

または

Dest, Src ;Comment

---

- Dest(「,」の左側部分)には【入力端子】を指定します。  
入力端子は、英数字3文字からなる端子の名前を指定します。
- Src(「,」の右側部分)には、Dest 部にて指定した入力端子に接続する相手となる【出力端子】または【数値】のいずれかを指定します。



出力端子の場合は、英数字3文字からなる端子の名前を指定します。

数値の場合は、-32768 ~ 32767の範囲の数値を記入します。

正の数を記載する際は符号は不要ですが、「+」をつけてもかまいません。

- Commentの部分には、自由に注釈文やメモなどを記述することができます。

行中にセミコロン「;」があると、そこから行末までの全ての文字が無視されます。

これを利用し、結線マクロの記述をわかりやすくするための説明を入れる事ができます。

※入力端子および出力端子の一覧は、本文書「▼全端子リスト」に、各機器の機能については「▼機器一覧」にそれぞれ記載しておりますので、併せて参照してください。

以上の記法による結線マクロの1行が、1本の結線情報に相当します。これを複数行にわたって組み合わせて、全体のプログラマブルタイマーの動作を定義していきます。

各行の記述を行う順序は不問ですが、同じ入力端子に関する行が重複して存在する場合は、後に出現した行が有効となります。詳しくは「▼記述競合時の処理」を参照してください。

#### ○コマンドの記法

コマンドは、プログラムの動作停止や再開、結線マクロのブロック開始・終了を宣言するために使用される命令語であり、結線情報の一部とは異なるものです。

コマンドについては、改行はCRコードまたはCR+LFコードです。

構文は下記の通りです。

---

#### Command

---

- Command部には、RUN,STOP,BLOCK0,BLOCK1,BLOCK2,BLOCK3,ENDBのいずれかが入ります。

RUNとSTOPについては、厳密に行頭からスペースを空けず記載してください。

それ以外については行頭のスペースやコメントの付加も可能です。

※各コマンドの機能や使い方は、本文書「▼コマンド一覧」に記載しております。

#### ▼二値信号と整数信号

出力端子の中には、0か1のいずれかの二値の信号を出力する端子と、本機が表現可能な範囲の整数である-32768 ~ 32767を信号として出力する端子とがあります。

また、入力端子の中でも、オフまたはオンのような2状態のいずれかを受け取る端子と、様々な整数値を受け取って動作する端子とがあります。

これらの二値の信号と整数の信号は、同種・異種問わず相互に端子接続可能です。

その際の信号の取り扱い方は下記の通りとなります。

○同種信号の取り扱い

- 二値の入力へ、二値の出力を接続する場合  
出力値の0、1がそのまま入力端子に0、1として入力されます。
- 整数の入力へ、整数の出力を接続する場合  
出力値の-32768 ~ 32767がそのまま入力端子に-32768 ~ 32767として入力されます。

○異種信号の取り扱い

- 二値の入力へ、整数の出力を接続する場合  
出力値が0の場合、入力端子には0が、  
出力値が0以外(-32768 ~ -1 または 1 ~ 32767)の場合、入力端子には1が入力されます。
- 整数の入力へ、二値の出力を接続する場合  
出力値の0、1がそのまま入力端子に(整数の)0、1として入力されます。

▼数値の種類制限

保存領域の容量上の制約のため、結線マクロの1つのブロック内に使用できる「整数値」は、最大「32種類」までとなります。同じ値が2箇所以上に使われる場合は1種類とカウントされます。

※0と1は頻出の値であるため、32種類の枠外で管理しておりカウントには含まれません。

32種類を超えて定義すると、シリアルポートのメッセージにエラー(コード WN01: Number table is full)を送信します。

結線マクロの構成を見直し、使用する数値の種類を減らしてください。

▼バージョン表記

各ブロック内には、作成した結線マクロのバージョンを埋め込むことができます。

結線マクロの新旧識別等に活用いただけます。

入力端子「MAJV」「MINV」が存在しますので、これらに対して整数値を接続する記述を、結線マクロと同じ要領でブロック内に記述できます。

作成した結線マクロの動作には一切影響しません。

MAJVはメジャーバージョン(大バージョン)、MINVはマイナーバージョン(小バージョン)を意味しますが、番号の付け方は自由です。MAJVとMINVのどちらか一方のみ使用しても、まったく使用しなくても問題ありません。

※MAJV,MINVは通常の入力端子と異なり、数値のみが接続できます。出力端子を接続するとエラーメッセージを送信します。

バージョン表記を付加したブロックの記述例は下記の通りです。この場合、結線マクロの保存時にVer. 1.2と表示されます。

---

**BLOCK0**

**MAJV, 1** ;メジャーバージョンは1です  
**MINV, 2** ;マイナーバージョンは2です →「Ver. 1.2」  
**RY0, TY0** ;以下、結線マクロの本体の記述が続きます  
:...

**ENDB**

---

※MAJV,MINVの使い方は一般の入力端子と同様の記法ですが、数値のみが接続できます。

出力端子を接続するとエラーメッセージを送信します。

※MAJV,MINVのいずれも未接続の場合、表示は「Ver. n/a」(not available=無し)となります。

**▼全端子リスト****【入力端子】**

○二値入力端子(0または0以外のどちらであるかを0、1として受け取る端子)

• RY0,RY1,RY2,RY3	リレー
• TP0,TP1,TP2,TP3,TP4,TP5,TP6,TP7	タイマー : 立ち上がりエッジトリガ
• TN0,TN1,TN2,TN3,TN4,TN5,TN6,TN	タイマー : 立ち下がりエッジトリガ
• TL0,TL1,TL2,TL3,TL4,TL5,TL6,TL7	タイマー : レベルトリガ
• TR0,TR1,TR2,TR3,TR4,TR5,TR6,TR7	タイマー : リセット
• NX0,NX1,NX2,NX3,NX4,NX5,NX6,NX7	NOTゲート : 入力
• DA0,DA1,DA2,DA3	ANDゲート : 入力a
• DB0,DB1,DB2,DB3	ANDゲート : 入力b
• DC0,DC1,DC2,DC3	ANDゲート : 入力c
• DD0,DD1,DD2,DD3	ANDゲート : 入力d
• DE0,DE1,DE2,DE3	ANDゲート : 入力e
• DF0,DF1,DF2,DF3	ANDゲート : 入力f
• DG0,DG1,DG2,DG3	ANDゲート : 入力g
• DH0,DH1,DH2,DH3	ANDゲート : 入力h
• WA0,WA1,WA2,WA3	ORゲート : 入力a
• WB0,WB1,WB2,WB3	ORゲート : 入力b
• WC0,WC1,WC2,WC3	ORゲート : 入力c
• WD0,WD1,WD2,WD3	ORゲート : 入力d
• WE0,WE1,WE2,WE3	ORゲート : 入力e
• WF0,WF1,WF2,WF3	ORゲート : 入力f

• WG0,WG1,WG2,WG3	ORゲート	: 入力g
• WH0,WH1,WH2,WH3	ORゲート	: 入力h
○整数入力端子(全範囲の整数値を受け取る端子)		
• JH0,JH1,JH2,JH3,JH4,JH5	ジャンパーセレクタ	: Hレベル入力
• JL0,JL1,JL2,JL3,JL4,JL5	ジャンパーセレクタ	: Lレベル入力
• TX0,TX1,TX2,TX3,TX4,TX5,TX6,TX7	タイマー	: カウンタ設定値
• TM0,TM1,TM2,TM3,TM4,TM5,TM6,TM7	タイマー	: カウンタ倍率値
• AA0,AA1,AA2,AA3	加算機	: 入力a
• AB0,AB1,AB2,AB3	加算機	: 入力b
• MA0,MA1,MA2,MA3	乗算機	: 入力a
• MB0,MB1,MB2,MB3	乗算機	: 入力b
• VM0,VM1,VM2,VM3	半固定抵抗	: 最大値

#### 【出力端子】

○二値出力端子(0または1のみを出力する端子)

• PPD	起動トリガ出力	
• PPC	常時1出力	
• PH0,PH1,PH2,PH3	フォトカプラ	
• JP0,JP1,JP2,JP3,JP4,JP5	ジャンパーレベル	
• TY0,TY1,TY2,TY3,TY4,TY5,TY6,TY7	タイマー	: 出力
• TD0,TD1,TD2,TD3,TD4,TD5,TD6,TD7	タイマー	: 遅延出力
• NY0,NY1,NY2,NY3,NY4,NY5,NY6,NY7	NOTゲート	: 出力
• DY0,DY1,DY2,DY3	ANDゲート	: 出力
• WY0,WY1,WY2,WY3	ORゲート	: 出力

○整数出力端子(-32768 ~ 32767の整数値を出力する端子)

• VR0,VR1,VR2,VR3	半固定抵抗	: 補正後出力値
• JY0,JY1,JY2,JY3,JY4,JY5	ジャンパーセレクタ	: 出力
• AY0,AY1,AY2,AY3	加算器	: 出力
• MY0,MY1,MY2,MY3	乗算機	: 出力

#### ▼機器一覧

本基板で使用可能な機器は、大まかに入力機器・出力機器・内部機器の3つに分類されます。

#### 【入力機器】

基板内へ情報を取り込むための機器として、フォトカプラ・ジャンパー・半固定抵抗が搭載されています。

結線マクロ上では、値を送出する端子(出力端子)をもった機器として扱われます。

○フォトカプラ 4個(部品名 PH1,PH2,PH3,PH4)

基板上のフォトカプラのオン・オフを取り込みます。

電圧の入力はネジ端子 CN6,CN7,CN8,CN9 に対して行います。

・[二値出力端子]PHn

フォトカプラへの電圧入力がない時 0、電圧入力がある時 1 を出力します。(n=0,1,2,3)

○ジャンパーレベル 6個(部品名 J1,J2,J3,J4,J5,J6・表示名 T0,T1,T2,T3,M0,M1)

ジャンパーピンに入力された電圧レベルを取り込みます。

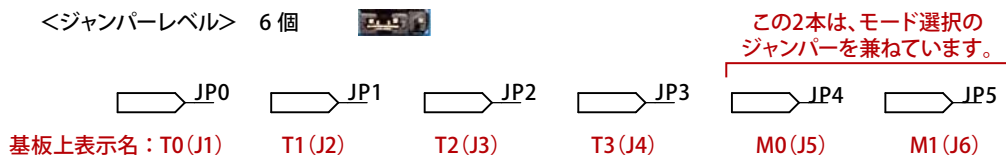
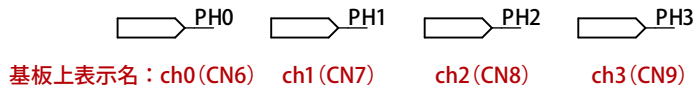
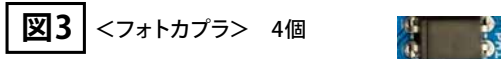
※外部論理入力としての使用法

ソケットをH側/L側どちらにも接続せず、中央ピンに直接外部からの5V TTLレベルの信号を入力することも可能です。保護回路はありませんので、電圧-0.5 ~ 5.5Vの範囲を絶対に超えないように注意してください。

※ジャンパーは2種の整数値をH/Lレベルに応じて切り替える内部機器「ジャンパー数値選択」としても使用できます。詳細は機器の項目を参照してください。

・[二値出力端子]JPn

ソケットをL側に接続した時0、H側に接続した時1が出力されます。(n=0,1,2,3,4,5)



○半固定抵抗 4個(部品名 VR1,VR2,VR3,VR4・表示名 V0,V1,V2,V3)

最大値を任意に設定可能な数値入力機器です。

・[整数入力端子]VMn

半固定抵抗を時計方向一杯まで回した時の最大値(※)を指定します。

数値は0以上のみ有効で、未接続または範囲外の数値の場合は1024が入力されたものとみなします。

・[整数出力端子]VRn

VMnの設定に基づいた出力値です。半固定抵抗のポジションに応じて、0 ~ VMn付近の値が出力されます。(n=0,1,2,3)

※ハード仕様により、半固定抵抗の分解能は1024段階となっています。

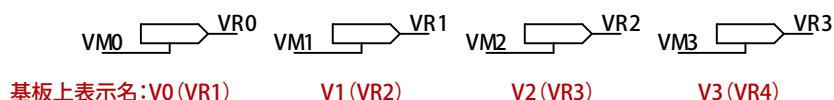
半固定抵抗の機械的な分解能は200分の1程度とされていますので、1024分の1の分解能は連続的な値を設定可能として扱って頂いても問題ありません。

一方、0も数値に含まれるため、半固定抵抗は0 ~ 1023の段階の値を取る事となります。

演算速度の都合から与えた値と半固定抵抗の値を乗算した後、1024で除算した答えの整数部をVRn出力としています。

このため、VMnを32767に設定した場合のVRnの最大値は $1023 \times 32767 \div 1024 = 32735$ となります(最小値は0です)。

**図4** <半固定抵抗> 4個



### 【出力機器】

情報を基板外部へ伝送する機器に、4個のリレーが搭載されています。

リレーは受け取った値に応じた動作をする機器ですので、信号を受け取る端子(入力端子)をもった機器として扱います。

○リレー 4個(部品名 RY1,RY2,RY3,RY4)

入力された信号に応じて接点の開閉状態が切り替わります。接点はネジ端子 CN2,CN3,CN4,CN5 につながっています。

・[二値入力端子]RYn

0を入力するとリレーがオフ(NO-COM：開、NC-COM：閉)、1を入力するとリレーがオン(NO-COM：閉、NC-COM：開)になります。(n=0,1,2,3)

**図5** <リレー> 4個



### 【内部機器】

基板上に部品として実在するものではなく、ソフトウェア上で仮想的に動作する機器の総称です。入力機器と出力機器の経路間に、タイマーを含めた各種の部品(内部機器)を組み合わせる事で複雑な機能を実現することができます。

○起動トリガ端子

・[二値出力端子]PPD

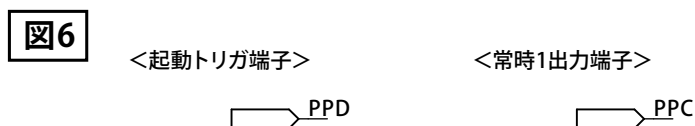
実行開始から100mS(0.1秒)経過後に立ち上がる信号を発生させます。

実行を開始した時点では0を、100mS経過後は1を出力します。

○常時1出力端子

- ・ [二値出力端子] PPC

実行中常に1を出力します。



○タイマー 8個

ソフトウェアで動作するリトリガブル型のタイマーが8個搭載されています。

機能の全体については本文書「▼タイマーの詳細」の項目を参照してください。

- ・ [二値入力端子] TPn

立ち上がりエッジトリガ：入力が0→1に変化した際にトリガ発生し、カウンタ値にTXnの入力値がセットされます。

未接続時は、この条件でのトリガは発生しません。

- ・ [二値入力端子] TNn

立ち下がりエッジトリガ：入力が1→0に変化した際にトリガ発生し、カウンタ値にTXnの入力値がセットされます。

未接続時は、この条件でのトリガは発生しません。

- ・ [二値入力端子] TLn

レベルエッジトリガ：入力が1の時にトリガ発生し、カウンタ値にTXnの入力値がセットされます。  
(この入力が1の間は常にトリガが発生し続けます)

未接続時は、この条件でのトリガは発生しません。

- ・ [二値入力端子] TRn

リセット：入力が1の時、カウンタ値が0にクリアされます。

(この入力が1の間はリセットがかかり続けるため、他のトリガが発生しても、カウンタ値は0を維持します)

- ・ [整数入力端子] TXn

カウンタ設定値：TPn, TNn, TLn 端子の各要因によってトリガ発生する際、カウンタにセットされる値です。

数値は0以上のみ有効で、未接続または範囲外の数値の場合は0が入力されたものとみなします。

- ・ [整数入力端子] TMn

カウンタ倍率値：タイマーの減算時間に倍率を掛けます。

数値は1～200の範囲のみ有効で、未接続または範囲外の数値の場合は1が入力されたものとみなします。

- ・ [二値出力端子] TYn

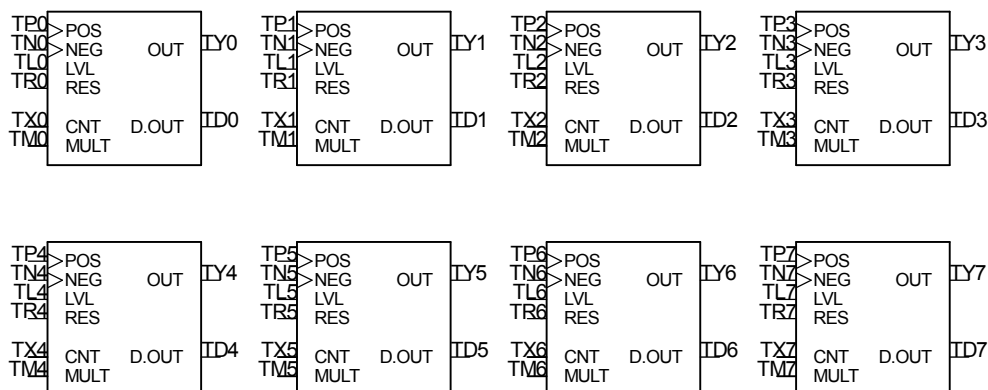
タイマー出力：カウンタ値が0の時(時間切れ)は0を、カウンタ値が1以上(走行中)の時は1を出力します。

• [二値出力端子]TDn

タイマー遅延出力：TYn端子の条件に加えて、カウンタ値が0になった直後の1ステップ間も1が出力されます。(n=0,1,2,3,4,5,6,7)

<タイマー> 8個

図7



○NOTゲート 8個

入力値の論理反転を行う論理ゲート素子(NOT、インバータ)を模した機器です。

• [二値入力端子]TXn

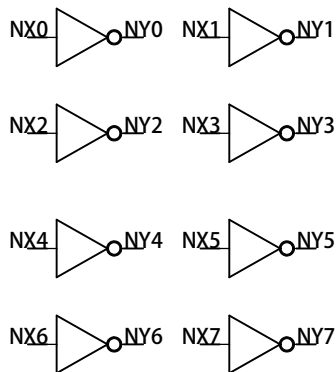
入力端子です。

• [二値出力端子]TYn

入力の反転が出力されます。入力が0ならば出力は1に、入力が1ならば出力は0になります。

(n=0,1,2,3,4,5,6,7)

図8 <NOTゲート> 8個

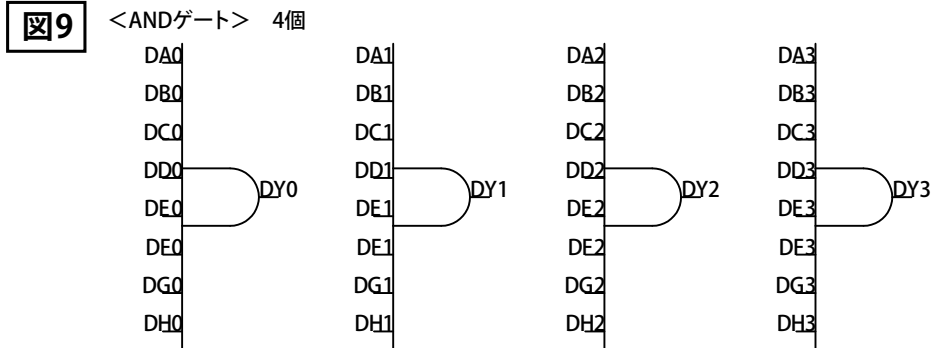




○ANDゲート 4個

最大8入力の論理積を取る論理ゲート素子(AND)を模した機器です。

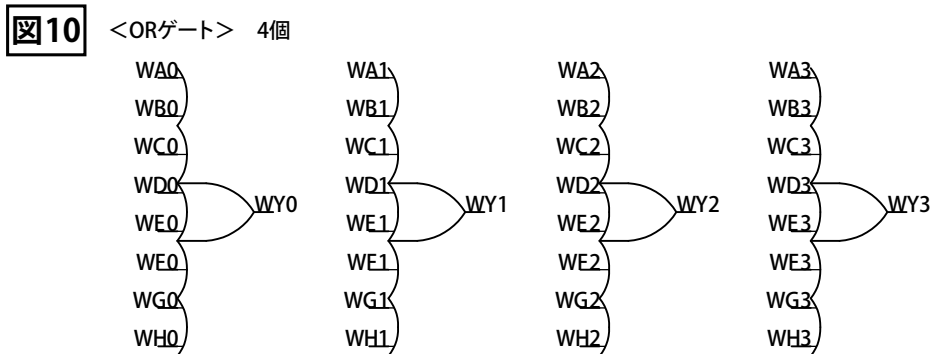
- [二値入力端子] DAn, DBn, DCn, DDn, DEn, DFn, DGn, DHn  
8本の入力端子です。未接続端子は1が入力されたものとみなします。
- [二値出力端子] DYn  
入力端子にすべて1が入力された場合 1、いずれか1本以上に0が入力された場合は0を出力します。(n=0,1,2,3)



○ORゲート 4個

最大8入力の論理和を取る論理ゲート素子(OR)を模した機器です。

- [二値入力端子] WAn, WBn, Wcn, Wdn, WEn, WFn, Wgn, WHn  
8本の入力端子です。未接続端子は0が入力されたものとみなします。
- [二値出力端子] WYn  
入力端子にすべて0が入力された場合 0、いずれか1本以上に1が入力された場合は1を出力します。(n=0,1,2,3)



○ジャンパーセクタ 6個(部品名 J1,J2,J3,J4,J5,J6・表示名 T0,T1,T2,T3,M0,M1)

ジャンパーの電圧レベル(H/L)に応じて、2種の数値から選択して出力するための内部機器です。

※H/Lレベルそのものを取得する場合は、入力機器「ジャンパーレベル」をお使いください。

・[整数入力端子]JHn

ジャンパーの電圧レベルがHの時に出力される数値です。

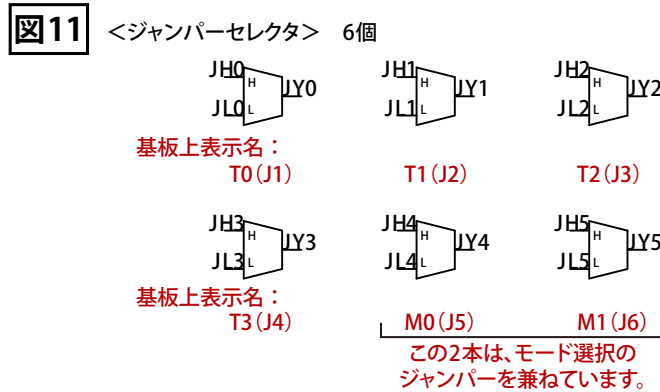
・[整数入力端子]JLn

ジャンパーの電圧レベルがLの時に出力される数値です。

・[整数出力端子]JYn

ジャンパーの電圧レベルがHの時はJHnの値を、Lの時はJLnの値が出力されます。

(n=0,1,2,3,4,5)



JHn：入力H ジャンパーがHの時に出力される数値を入力します。

JLn：入力L ジャンパーがLの時に出力される数値を入力します。

JYn：出力

ジャンパーがHの時：JHnの値

ジャンパーがLの時：JLnの値

出力されます。

○加算器 4個

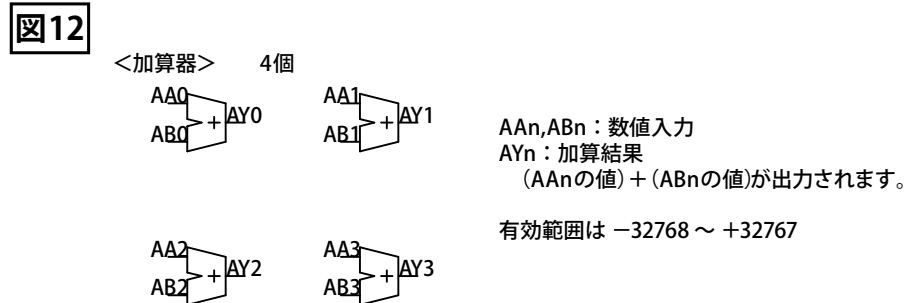
入力された2値の和(足し算)を求めて出力する機器です。

・[整数入力端子]AA<sub>n</sub>,AB<sub>n</sub>

加算する2つの数値です。

・[整数出力端子]AY<sub>n</sub>

加算結果(AAnの値)+(ABnの値)が出力されます。結果が-32768～32767の範囲を超える場合は不正な値が出力されますのでご注意ください。(n=0,1,2,3)



### ○乗算器 4個

入力された2値の積(掛け算)を求めて出力する機器です。

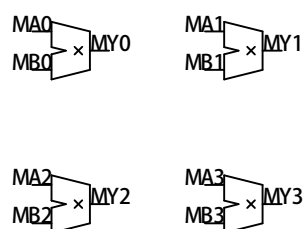
- [整数入力端子] MAn, MBn

乗算する2つの数値です。

- [整数出力端子] MYn

乗算結果(MAnの値)×(MBnの値)が出力されます。結果が-32768 ~ 32767の範囲を超える場合は不正な値が出力されますのでご注意ください。(n=0,1,2,3)

**図13** <乗算器> 4個



MAn, MBn : 数値入力  
MYn : 加算結果  
(MAnの値) × (MBnの値)が出力されます。

有効範囲は -32768 ~ +32767

### ▼動作状態

本基板は、下記に示す3種の状態を取ります。

電源投入時およびリセット時は「実行状態」から始まります。

#### ○実行状態 (Run)

結線マクロで定義されたプログラマブルタイマーの動作を実行中の状態です。

STOPコマンドによって明示的に停止させるまでは、永続的にこの状態を続けます。

この状態では、STOPコマンドを除く全てのコマンドと結線マクロを受け付けません。

#### ○停止状態 (Stop)

プログラマブルタイマーの動作を停止し、結線マクロおよびコマンドの転送を受け付けている状態です。

BLOCK0 ~ 3いずれかのコマンドを受信すると、結線マクロの収集を開始します。

順次送られてきた結線情報の文法を解析し、受理もしくはエラーを出力します。

ENDBコマンドを受信すると、ブロック内の結線マクロを確定し、保存のためUpdate状態に移行します。

#### ○停止状態で結線マクロの書き込み中(Update)

受信完了した1ブロック分の結線マクロを基板上マイコンの不揮発性メモリに保存している状態です。保存には最大で数秒程度の時間がかかります。完了次第、Stop状態に戻ります。

## ▼コマンド一覧

結線マクロ転送のコントロールに使用するコマンド単語の一覧を記載します。

### STOP

プログラマブルタイマーの実行を停止し、結線マクロおよびコマンドを受け付けます。

※この命令は行頭に空白を開けずに記述する必要があります。

### RUN

ジャンパーM1,M0で指定された動作モードの結線マクロに基づき、プログラマブルタイマーの動作の実行を開始します。

※この命令は行頭に空白を開けずに記述する必要があります。

### BLOCK0

### BLOCK1

### BLOCK2

### BLOCK3

動作モードの番号0～3に対応する結線マクロのブロック開始を宣言します。

### ENDB

結線マクロのブロック終了を宣言します。

これを受信することで、BLOCK0～3のコマンド受信以後に受け取った結線マクロの内容を確定し、基板上マイコンの不揮発性メモリへ保存します。

※BLOCK0～3とENDBは必ず対になるように使用してください。

ENDBコマンドが受信される前に、新たなBLOCK0～3いずれかのコマンドが受信された場合は、これまでに収集された結線マクロ内容を保存せずに破棄した上で、新たなブロックの結線マクロを受け付けます。

※ENDBの受信後は、不揮発性メモリへの書き込み処理を行うため最大数秒の時間がかかります。この間にはほかのコマンドや結線マクロを送信しないでください。

#### ▼エラーメッセージ一覧

結線マクロの入力時、文法エラーや受理できない接続指示が発生した場合は、エラーメッセージを出力します。

##### ○エラーコード PT01: Syntax Error "....."

数値や名称のいずれかが区別できない単語や、本マクロで使用しない文字(英数字と半角記号「+」「-」「,」「;」以外)が含まれているため、解釈できません。

全角文字や全角スペースが含まれている、もしくはコメントの前方に「;」を付け忘れている場合がありますので確認してください。

##### ○エラーコード PI01: Undefined identifier "....."

本プログラムで定義されていない端子名またはコマンド名です。

本文書「▼機器一覧」「▼コマンド一覧」を参照し、名前を確認の上修正してください。

##### ○エラーコード PN01: Number overflow

整数値の指定可能範囲を超えています。-32768 ~ 32767を指定してください。

##### ○エラーコード PN02: Bad sign in number

整数値の符号に不正な入力があります。符号は数値の前方に1つだけつけることができます。

##### ○エラーコード PF01: Expected COMMAND or INPUT device

行頭に(コマンド名または入力端子名)以外のものがあります。

行頭にはコマンド名もしくは入力端子名が必要です。

##### ○エラーコード PF02: Excess parameter

構文の後ろに不要な文字列があります。

コマンド行の場合は、コマンド名の後ろに「,」や他の文字列がないことを、結線行の場合は「入力,出力」の構文通りであることを確認してください。

##### ○エラーコード PF03: Expected comma (" , ")

結線行の入力端子名に続くカンマ「,」がありません。

入力端子名と出力信号(出力端子名または数値)の間は、「,」で分離してください。

##### ○エラーコード PF04: Expected OUTPUT device or NUMBER

結線行の「,」に続く出力指定に誤りがあります。「,」の後は、入力端子に接続する出力端子名または数値を記述してください。

○エラーコード WB01: Statement is out of block (first line here)

ブロック開始が宣言されていない状態で結線マクロが出現しました。

結線行の前にBLOCK0～3コマンドで、ブロックの開始を宣言してください。

または、想定していない ENDB コマンドでブロックを閉じてしまっていないか確認してください。

※複数行にわたる場合は、最初の1行で一度のみエラーを表示します。

○エラーコード WT01: INPUT devices cannot be declared as a source

出力信号の指定箇所(「,」の後部)に入力端子を記述しています。

信号を出力する端子名を確認の上、修正してください。

○エラーコード WT02: OUTPUT devices cannot be declared as a destination

入力端子の指定箇所(「,」の前部)に出力端子を記述しています。

信号の入力先となる端子名を確認の上、修正してください。

○エラーコード WN01: Number table is full

整数値を保存するための領域がいっぱいです。

ブロック内に出現する整数の値の種類が多いため、これ以上保存できません。

(0と1を除く整数値の使用可能な種類には上限があり、各ブロック内32種までです)

○エラーコード WV01: Only numbers can be accepted for the VERSION section

バージョン番号用の入力端子(MAJV,MINV)に整数値以外を接続する記述があります。

MAJV および MINVには、それぞれ整数値のみが接続できます。

▼記述競合時の処理

入力端子に接続できる出力端子は1本のみとなります。同一の入力機器に関する結線マクロの行を複数回追記した場合、最後に出現(最後に記述)した出力が優先されます。

先にあった出力は後の出現で上書きされ、無効となります。

例えばフォトカプラ0とフォトカプラ1を同じリレー0に接続するマクロ

---

RY0, PH0 ; フォトカプラ 0 をリレー 0 に接続

RY0, PH1 ; フォトカプラ 1 をリレー 0 に接続

---

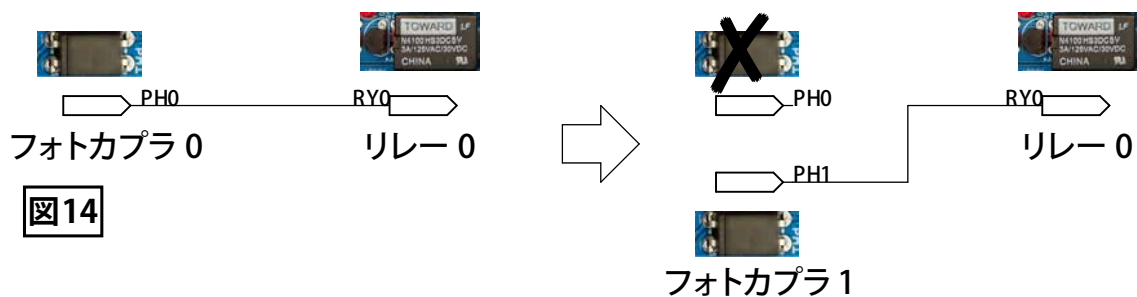
上記の場合、結線マクロの転送時に後から出現した「RY0,PH1」が有効となります。

※原則として同一ブロック内における結線マクロの記述順は不問ですが、このように同じ入力

端子に関する行が重複する場合には、後に出現する行の記述が有効となるため、マクロの

記述順が動作に影響を与えることとなります。意識して作成する場合を除き、同じ入力に複数

の出力を接続する結線マクロは混乱を回避する点からお勧めしません。



逆に出力は無制限に複数の入力端子に対して接続する事ができます。

(出力のコピーはいくらでもOK)

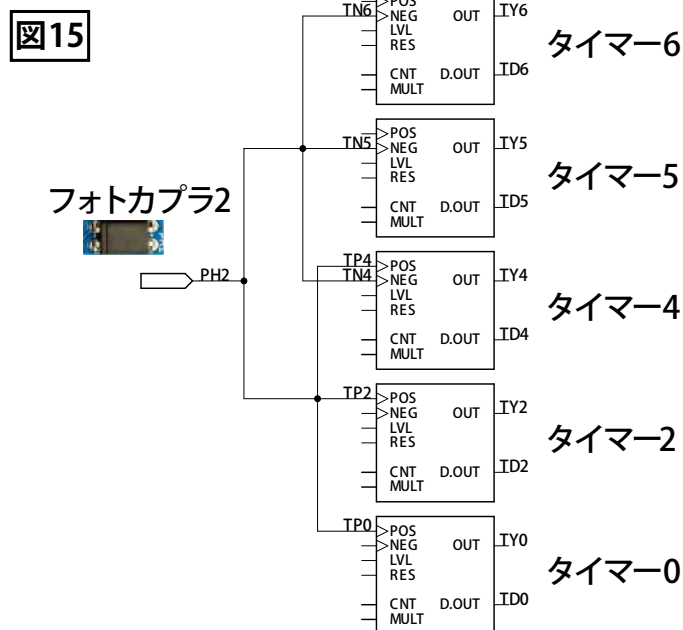
例えばフォトカプラ2番でタイマー0,2,4を立ち下がりエッジトリガ、4,5,6を立ち上がりエッジトリガする場合、

(タイマー4は二重で、立ち上がりと立ち下りの両方のトリガを受付)

結線マクロは下記のようになります。

- TP0, PH2 ; タイマー0をフォトカプラ2の入カレベル0→1へのエッジトリガ
- TP2, PH2 ; タイマー2をフォトカプラ2の入カレベル0→1へのエッジトリガ
- TN4, PH2 ; タイマー4をフォトカプラ2の入カレベル1→0へのエッジトリガ
- TP4, PH2 ; タイマー4をフォトカプラ2の入カレベル0→1へのエッジトリガ
- TN5, PH2 ; タイマー5をフォトカプラ2の入カレベル1→0へのエッジトリガ
- TN6, PH2 ; タイマー6をフォトカプラ2の入カレベル1→0へのエッジトリガ

この例ではフォトカプラ2番に信号が入ればタイマー0,2,4がスタートし、信号が無くなればタイマー4,5,6がスタートします。



#### ▼プログラムの内部動作について

本基板のプログラマブルタイマーは、結線マクロとして定義された結線情報にもとづく入出力の処理と、タイマーの進行処理を行います。

処理のステップは10mS(0.01秒)間隔、すなわち1秒あたり100ステップの割合で行われます。

各ステップでは、下記の順序で処理が実行されます。

#### 1. 基板上入力端子の読み取り

- ・ フォトカプラの読み取り ×4個
- ・ ジャンパー電圧レベルの読み取り ×6個
- ・ 半固定抵抗ポジションの読み取り ×4個

#### 2. 全機器の出力決定

内部機器全ての出力状態を、結線情報にもとづく入出力関係をたどりながら漸次的に決定していきます。

下記に示す順序で、全体を8回反復して決定します。同種の機器は、番号が若いものから順番に処理されます。

2-1. NOTゲート(0,1,2,3,4,5,6,7番の順)

2-2. ANDゲート(0,1,2,3番の順)

2-3. ORゲート(0,1,2,3番の順)

2-4. 半固定抵抗：最大値設定によって補正された出力値(0,1,2,3番の順)

2-5. 乗算器(0,1,2,3番の順)

2-6. 加算器(0,1,2,3番の順)

2-7. ジャンパー数値選択(0,1,2,3,4,5番の順)

#### 3. タイマーの処理

下記手順を一まとめとして、0,1,2,3,4,5,6,7番の順に処理します。

3-1. 各種トリガ、リセットトリガ有無の判定

3-2. カウンタの減算処理

3-3. 出力の決定

#### 4. リレー出力への反映 ×4個

#### ▼スキューの回避

タイマーの入出力処理の内部では、同種の機器は番号が若いものから順番に出力端子の確定処理を行っていきます。

例えば、NOTゲート(0を入れると1が出力され、1を入れると0が出力される)を3個直列に接続する場合を考えます。



#### 例 1：昇順に使用

フォトカプラ0番の入力を、NOTゲートの0番、1番、2番の順に通し、結果をNOTゲート2番から取り出す場合、結線マクロは下記ようになります。

---

NX0, PH0 ; フォトカプラ0をNOT0に接続  
NX1, NY0 ; NOT0の出力をNOT1に接続  
NX2, NY1 ; NOT1の出力をNOT2に接続

---

NOTゲートの出力状態は0番から順に判定される規則のため、NOT0の状態は、フォトカプラ0で決定します。次にNOT1の出力は、NOT0の出力が確定済のため確定可能です。同様にNOT2の出力も、NOT1が確定済のため確定可能です。0～2番すべてのNOTゲートについて、出力判定を行う時点で、既に入力が確定しているため、1度の繰り返しで正しい結果を求めることができます。

#### 例 2：降順に使用

一方、これを逆にNOTゲートの2番、1番、0番の順に接続すると、結線マクロは下記ようになります。

---

NX2, PH0 ; フォトカプラ0をNOT2に接続  
NX1, NY2 ; NOT2の出力をNOT1に接続  
NX0, NY1 ; NOT1の出力をNOT0に接続

---

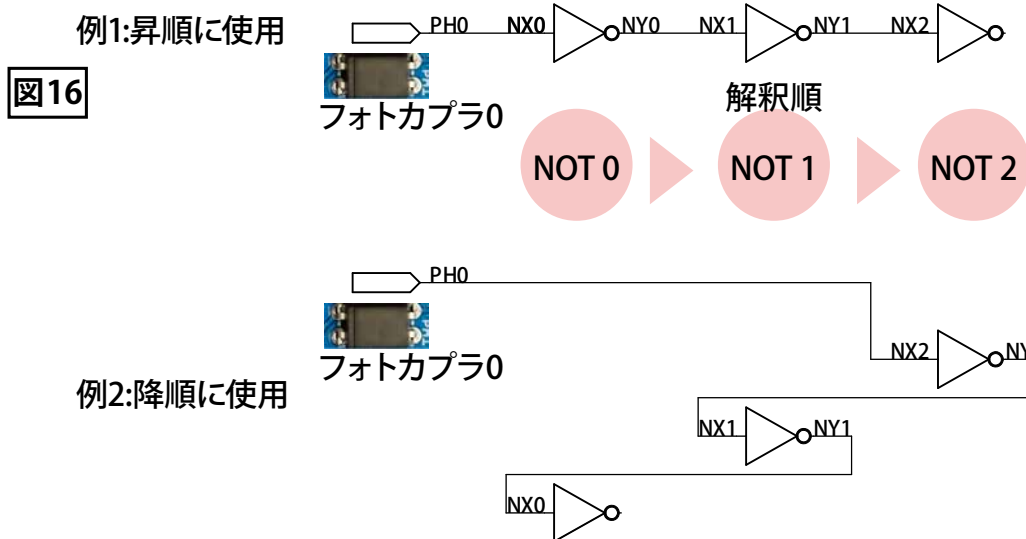
論理的には正しいのですが、NOT0を判定した時点ではまだNOT1の出力は確定されていなくて、以前の値を保持している状態です。NOT1についても同様に、NOT2の出力は古いままです。

動作的には、

- ・ 繰り返し1回目：NOT2にフォトカプラの入力が反映される。NOT0,1は古い値をもとに動作中
- ・ 繰り返し2回目：NOT1にNOT2の情報が反映される。NOT0は依然古い値で動作中
- ・ 繰り返し3回目：NOT0にNOT1の情報が反映される

都合3回の解釈を以って正しい状態に落ち着きます。上の例1が一回の繰り返しで最終結果にたどり着くのに対し、この例2では遅れが出ます。

単純な動作ではこの遅れは無視できますが、反転の途中から他の素子に繋がっている等の組み合わせがあった場合時間的に不一致になる瞬間が発生します。



こうした判定順序の時間遅延(スキュー)によって、正常値が得られない期間が発生するのを極力回避するため、上記の各出力値の判定処理を1ステップ(10mS間隔)あたり8回反復するようになっています。

これにより、8段以内の時間差発生は起こりません。そのため、例1(所要1段)と例2(所要3段)では同じ結果が得られます。

しかしながら、9段以上の解釈順が逆になる結線状態は上記の現象が発生する事にご留意ください。

(通常はこのような複雑な結線状態は無理に作らないと存在しませんが、理論上は発生します) 各機器の出力判定の順序は、本文書「▼プログラムの内部動作について」をご確認ください。

#### ▼タイマーの詳細

本機が搭載するソフトウェアタイマーは、1ステップ=10mS(0.01秒、100Hz)をクロックとしてダウンカウントを行います。

セットしたカウンタ値が1以上(走行中)の時に出力がオン、0になると(停止中)出力がオフになります。

タイマーにセットできるカウンタ値の最大は32767です。

32767をセットした際のタイマー時間は $32767 \times 0.01 = 327.67$ 秒(約5分27秒)になります。

また、長周期のタイマー時間が必要な場合に備えて倍率設定を用意しています。

この入力には1～200の数値を指定する事ができ、クロックをこの数値分だけ引き伸ばす事ができます。

例えば50を指定すると、カウント周期が50倍遅くなり、タイマー時間が50倍長くなります。

指定できる最大値の200では10mS周期が200倍されて2000mS(2秒)周期になり、最大タイマー時間は65534秒(約18.2時間)になります。

タイマーには3つのトリガ入力があります。トリガ入力はタイマーに時間を設定(入力端子名 TXn にて指定するカウンタ値でタイマーをスタート)します。(n=0,1,2,3,4,5,6,7)

・立ち上がりエッジトリガ(入力端子名 TPn)

0から1への変化点でタイマー値が読み込まれます。1が続いても読み込みは発生しません。一度0に戻った後に1になると読み込みが発生します。

・立ち下がりエッジトリガ(入力端子名 TNn)

1から0への変化点でタイマー値が読み込まれます。0が続いても読み込みは発生しません。一度1に戻った後に0になると読み込みが発生します。

・レベルトリガ(入力端子名 TLn)

入力が1になっている間タイマー値が読み込続けられます。入力が0になるまで読み込みを続けますので、入力が1になっている間はタイマー時間が進みません。

これら3本の入力は同時に使用する事もできます。また結線されなかったトリガ入力は無かったものとして扱われます。

レベルトリガは、入力が1になっている間、カウントが停止状態であるのに比べ、エッジトリガは変化点のみを捉えて数値を設定するため、ある瞬間から指定時間によるタイマーをスタートする事ができます。

レベルトリガは何かの事象が有効な間とその後の一定時間が必要な用途に使用します。

例えば人体感知素子(PIRセンサー)と組み合わせ、人が存在している間と、人が居なくなっから一定時間電灯を点灯させるような用途に使用します。

○エッジトリガの動作モード

タイマーのトリガ動作モードはリトリガブルです。リトリガブルとは、トリガが発生して時間カウントが進んでいる途中でも、新たなトリガが発生すると最初から時間カウントをやり直す手法です。

○リセット入力

1を入力した場合、現在進行中のカウント値は0になり、出力がOFFになります。

0に戻るまで、いかなるトリガ入力も無視します(トリガより優先)

タイマー時間を0にしてトリガを発生させると、時間に0が読み込まれます(出力はオフ(0)になる)これはリセット入力を1にしたのと同じ状態です。

○遅延出力

タイマー出力を次のタイマーのトリガとする順次接続、例えば下記の結線マクロ

---

TN1, TY0 ; タイマー0の出力をタイマー1の立ち下がりエッジに接続

---

この結線は、タイマー0の動作終了(=時間切れ、出力TY0が1→0になる)で次のタイマー1をトリガしてスタートさせています。

この動作を動作ステップ(10mS単位)で見ていった場合、次の因果関係があります。

(詳細な内部動作順序「▼プログラムの内部動作について」参照)

- ・タイマー動作はトリガが発生した事をチェックする。
- ・カウント値が0でなければ-1する。
- ・出力値を決定する(カウント値=0ならば0、1以上ならば1)。

タイマー0が時間切れになると、タイマー1をトリガしますが、タイマー1のトリガ条件を調べるのが先で、タイマー0の出力が調べられるのが後になります。

このため、トリガ動作は次の周期に持ち越されます。

すなわち上記動作には1周期分時間が必要です。

これは以下の処理手順を実行しているソフトウェアタイマーの仕様です。

タイマーの出力検査はほぼ最下位にありますが、トリガのような論理値は優先順位が高いため、タイマー確認が終わった次ぎの周期の頭でトリガ条件が整えられます。

結果として、タイマー終わりから他のタイマーへのトリガでは1周期分のブランク期間が生じます。

縦列接続したタイマーが全て停止した事を確認したい場合は、OR機能(論理和機能)を使ってタイマー出力を集合させれば良いのですが、1周期の不動期間が存在してしまいます。

これを避けるため、タイマーには1クロック分の遅延時間を持った出力を用意しています。

この出力は通常の出力が1になるタイミングと同じタイミングで1になり、通常の出力が0に戻った1ステップ後に0に戻ります。(立ち上がりは同じで立下りだけ1ステップ分の遅延)

縦列タイマーが全て終了した事を検出する場合はこの遅延出力TDnをご利用ください。

出荷時にセットしているMode3の結線マクロ(縦列タイマー)で、実際に遅延出力端子を利用しています。

#### ○動作開始時の立ち上がりエッジトリガ

タイマーの両エッジトリガ端子(TPn, TNn)は、動作開始時には全て0が入力されている状態から開始します。

立ち上がりエッジトリガの入力に数値の1を結線した場合、結線マクロでは下記のようになります。

---

TP0, 1 ; タイマー0の立ち上がりエッジトリガ入力を1に固定

---

この場合、動作開始直後の入力端子レベルを0と解釈しますので、最初のステップで0→1に変化するため立ち上がりのエッジトリガが発生します。

(起動直後に1度だけ発生する「起動トリガ」として機能します)

この他、起動直後のワンショットトリガとして使用できる出力端子に、起動開始後100mS(0.1秒)に0→1になる信号(出力端子名 PPD)を用意していますのでそちらの利用もご検討ください。