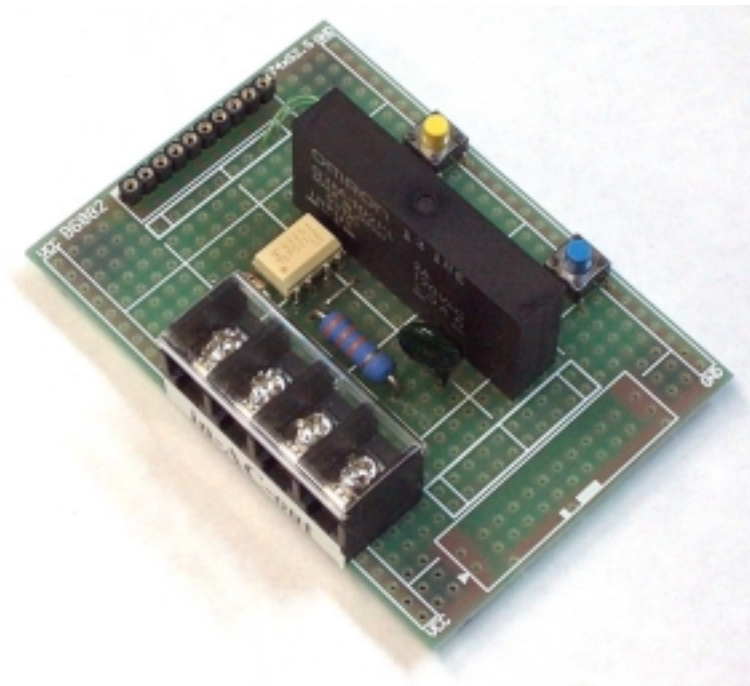


- TK-3687 Option -

# AC パワーコントローラ

Version 1.03



---

---

## 目次

---

---

1・はじめに、、、	1
2・位相制御とは	2
3・システム設計	3
4・ハードウェアの設計	5
5・ソフトウェアの設計	7
6・ハードの製作と動作チェック	12
7・付録	16

---

## 1 はじめに、、、

このキットはソリッドステートリレー(SSR)とフォトカプラを使用した AC パワーコントロールキットです。フォトカプラでAC100Vのゼロクロス(AC電源の0Vの時)を検出し位相制御でSSRをon/offし出力をコントロールします。パワー制御を行うには幾つかの方法がありますが、その一つが位相制御方式です。この位相制御はハードが簡単で滑らかな制御ができ、マイコン制御に向いている為よく用いられます。

ここでは通常の開発行程と同様に位相制御の解析から始まり、システム設計、ハードウェアの設計、ソフトウェアの設計、製作及び動作チェックと続きます。しかし、先に作りたい方は製作してからでも良いかもしれません。システムを動作させ自分なりのイメージを持つことが出来ればテキストの理解もし易いと思います。マイコンの仕組みやハードとソフトの関係を調べ、実用的なレベルまでスキルアップするには相当な時間と労力を費やしますが、本キットでその足掛かりを掴んで下さい。また、フロアスタンドや電気ドリルなど身の回りの器具にも応用できますので実用性もあります。

作業を進めていくに当たり次のものを用意して下さい。

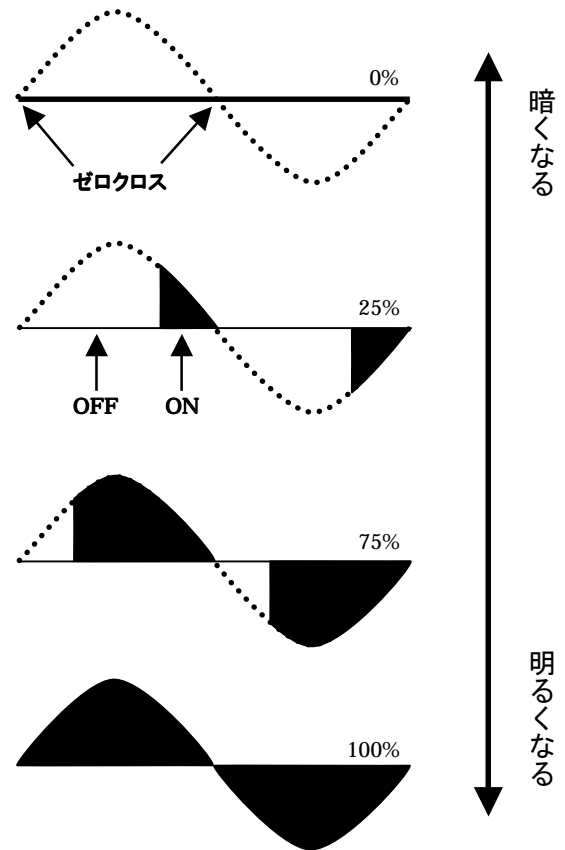
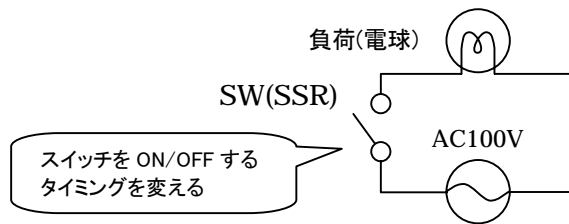
- ・**組み立て**      ハンダごて、ハンダ、ニッパ、ワイヤストリッパ、ピンセット、+ドライバ、テスタ
- ・**動作確認**      ハイパーターミナル(Windows 付属)、D-Sub9pin ストレートケーブル
- ・**プログラム**      HEW(High-performance Embedded Workshop)、もしくは Motorola 形式 HEX ファイル(\*.mot) を生成するアセンブラ(本文内で掲載しているリストは HEW を使用して作成しています)

では、まず位相制御とはどのようなものなのか調べていくことにしましょう。

## 2 位相制御とは

位相制御とは、負荷と AC 電源の間にスイッチを設け、ゼロクロスにスイッチを off し、そこからある一定の時間(ゼロクロスから次のゼロクロスまでの半周期の間のどこか)が来たらスイッチを on して負荷に電圧を加えるという動作を半周期毎に毎回行うことによって負荷に加わるパワーをコントロールする方式です(図 2-1)。

AC スイッチには電子的なスイッチであるソリッドステートリレー(SSR)を使用します。制御側ではゼロクロス検出部からの信号を基準として、そこから出力する制御量によって予め設定された時間になったら SSR を駆動して AC100V を on します。操作スイッチに応じゼロクロスから SSR を駆動するまでの時間をマイコンにより徐々に長くしたり短くしたりすることにより、照明の場合ならば段々と明るくしたり暗くしたりする事が可能になります。



< 図 2-1 位相制御 >

### 3 システム設計

詳細設計に入る前にまずシステム全体の構成を考えて見ましょう。今回のシステムは3つの部分、つまり、

- ①位相制御部(中心となる I/O)
- ②操作スイッチ部(ユーザから指示を受ける)
- ③表示部(ユーザに動作状況を知らせる)

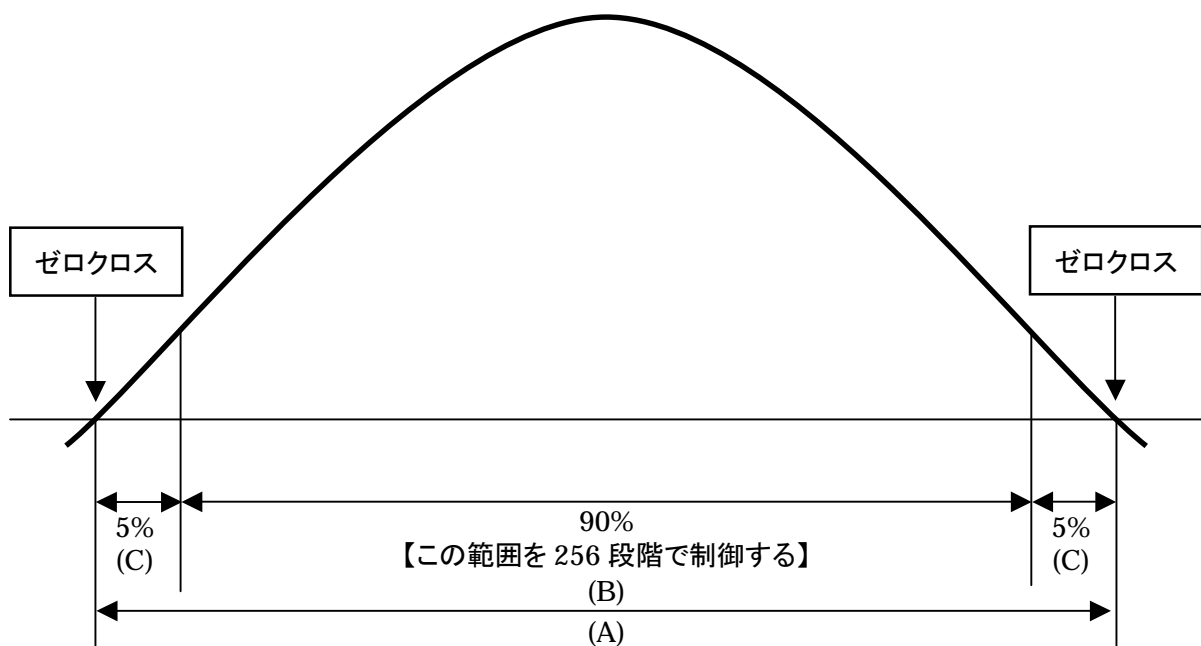
に分けることができます。それぞれの部分について検討してみましょう。

#### ①位相制御部

位相制御とは、冒頭でも述べた通り負荷とAC電源の間にスイッチを設け、ゼロクロス時にスイッチをoffし、そこからある一定の時間(ゼロクロスから次のゼロクロスまでの半周期の間のどこか)が来たらスイッチをonして負荷に電圧を加えるという動作を半周期毎に毎回行うことによって負荷に加わるパワーをコントロールする方式です。負荷に加わるパワーは波形の面積に当たりますから、ゼロクロス近辺では多少onする時間を変化してもパワーはあまり変わらないことがわかります。ということは、ゼロクロス近辺で細かく制御するよりも、波形の中心付近で細かく制御する方が制御的には良いといえるでしょう。そこで、制御範囲はゼロクロスから次のゼロクロスまでの間(電源周期の半分)を100%とした時、ゼロクロスから5%の時間が過ぎたところから、次のゼロクロスの5%の時間前までの90%の間とします。この90%に相当する時間を分割し、パワーを変更するときはゼロクロス毎に1段階ずつ増加、もしくは減少します。分割数は256段階とします。もちろんパワー0の時は出力を全く出さず、フルパワー時はAC100Vをそのまま出力したいので、0段階(最小値)の時には完全に出力をoffし、255段階(最大値)の時には出力をonします。

尚、今述べた割合と時間の関係は電源周波数(50/60Hz)によって変更が必要です。せっかくマイコンで制御するので、周波数による切り替えもソフトウェアで行いましょう。判別の方法はゼロクロスの間隔を何回か計測し、その平均値が50Hzと60Hzのどちらに近いかによって行います。

以上の関係を図3-1に、それぞれの時間を表3-1に示します。



<図 3-1 制御区間>

周波数	(A)	(B)	(C)	(B)/256
50Hz	10.0msec	9.0msec	0.5msec	35.16 $\mu$ sec
60Hz	8.333msec	7.500msec	0.417msec	29.30 $\mu$ sec

<表 3-1 周波数毎の各時間>

この時間管理は CPU 内蔵の CTC・タイマ Z を使用します。タイマ Z は 16 ビットカウンタですので msec オーダーのカウンタが可能です。また、外部からの信号でのフラグのセットやカウント値をレジスタにセット機能が備わっています。今回はこの機能を利用して位相制御を行うことにします。

## ②操作スイッチ部

出力設定は Up/Down スイッチで行う事とし、各スイッチの機能は次のようにします。

Up	押されている間出力アップ、離すとそこで停止
Down	押されている間出力ダウン、離すとそこで停止

尚スイッチが同時に押された場合、Up スイッチを優先して受け付けます。

## ③表示部

制御量の表示は CPU 基板のポート 5 に接続されている LED を使用します。最小値(出力 off)の時に消灯、出力が増えるにつれ LED を 1 ビットずつ点灯してゆき最大出力の時は全点灯と最大最小を含めた 9 段階で表示します。

出力	点灯パターン
Off	
小	
↑	
↓	}
大	
On	

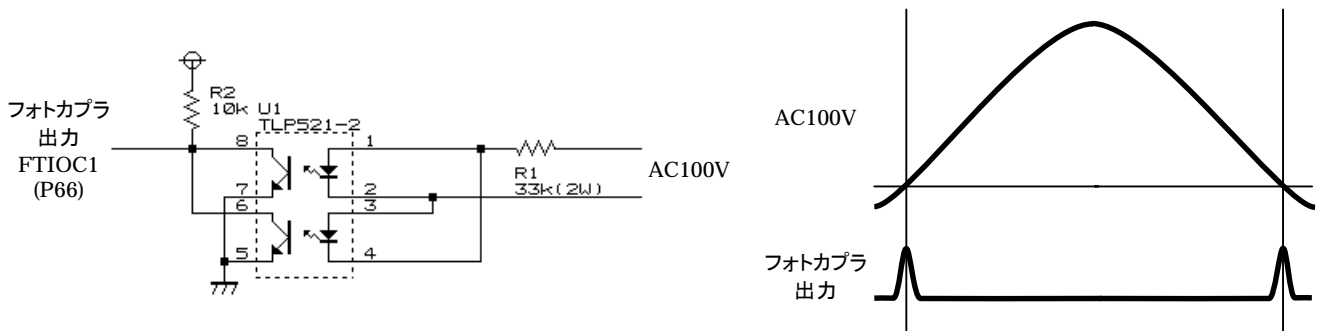
<表 3-2 出力と点灯パターン>

## 4 ハードウェアの設計

では、各ブロック毎にハードウェアの設計をしましょう。

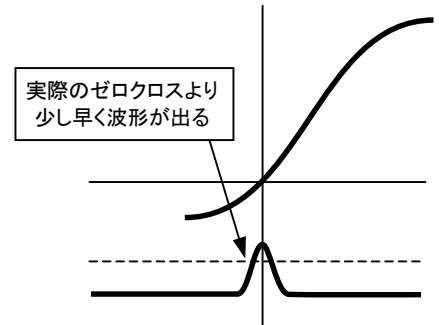
### ①位相制御部(ゼロクロス検出・SSR 制御)の設計

位相制御方式のタイミングはゼロクロスが基準となりますので、AC100V のゼロクロスを検出してその信号を CPU に入力します。ゼロクロス検出には 2 回路入りのフォトカプラを使い、全波整流・絶縁・ロジックレベルの変換を一度に行います。このフォトカプラの出力をタイマ Z の FTIOC1 (P66)に入力しゼロクロスを検出します。フォトカプラの出力はかなり鈍っていますが CPU の内部回路にシュミットトリガのゲートが入っているので問題ありません。また、検出はレベルではなくエッジですのでそのまま入力して構いません。検出回路とフォトカプラ出力タイミングを示します。



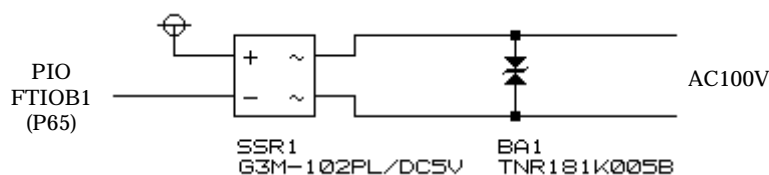
＜図 3-2 ゼロクロス検出回路とフォトカプラ出力タイミング＞

図 3-3 はフォトカプラ出力時の波形を拡大したものです。破線は検出電圧を示します。図から読み取れるように CPU は実際のゼロクロスより手前で検出してしまいます。時間的には完全に 0V の時に信号が出るのが理想ですが、これをハードウェアで行うと少し複雑になってしまいます。それに 0V の時に検出が出来たとしてもプログラムの実行に時間がかかるので、結局ゼロクロスからずれてしまいます。以上のことからこの辺の調整はソフトウェアで行うことにします。



＜図 3-3 ゼロクロスの検出と実際＞

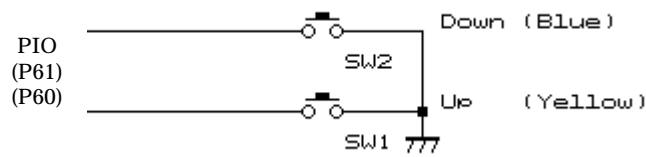
次に AC100V の on/off を制御する SSR の回路を考えてみましょう。On/off の時間は内蔵タイマを使用するのでハードでは特に考慮する必要はありません、純粹にポート出力で SSR をコントロールします。ポートは FTIOB1 (P65) を出力として使用します。尚、CPU 基板の各ポートにはプルアップ抵抗が接続されているので、電源 on でいきなりフルパワーが出力されることはありません。AC100V 側にはサージ電圧吸収用にバリスタを接続します。これはモータなど誘導性負荷接続時に生じるサージ電圧から SSR を守る為です。尚、ここで使用している SSR は 2A まで電流を流せますので接続できる容量は 200W までです。回路図を以下に示します。



＜図 3-4 SSR 制御回路＞

## ②操作スイッチ部

操作スイッチはUpスイッチ・Downスイッチの二つです。チャタリング除去はソフトで行うので単純にポートとスイッチを接続します。接続ポートは位相制御で上位4ビットを使用しているの下位4ビットのP60、P61とします。プルアップはCPU基板上で施されているのでプルアップ抵抗は不要です。回路図を示します。



<図 3-4 操作スイッチ回路>

## ③表示部

出力レベルのモニタ表示はCPU基板・ポート5のLEDを使用するので設計は不要です。



以上でハードの設計は終了です。次はソフトウェアの設計に入ります。尚、ここで先にハードの製作を行ってしまっても構いません。ハードの製作は5章で説明していますのでそちらを参照して下さい。

## 5 ソフトウェアの設計

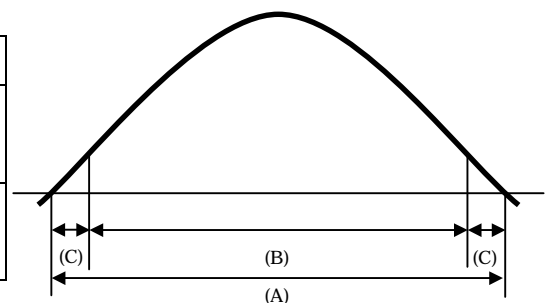
回路が完成したら、次はこれを制御するソフトを作成します。2章のシステム設計でタイマZを使用する事は決まっていますので、ソフトはこのタイマZを利用して位相制御を行います。それでは各制御を考えましょう。

### ①位相制御部(タイマZの設定)

タイマZにはインクリメントカウンタと4つのジェネラルレジスタがあり、カウンタの値とジェネラルレジスタを常に比較しカウンタ値がジェネラルレジスタの値と一致したら出力を変化させる“コンペアマッチ”機能があります。今回はこの機能を使って位相制御を行います。タイマZの初期出力をHi、つまりSSRがOffにセットしておき、コンペアマッチが発生したらLowを出力するように設定します。SSRにはFTIOB1が接続されているのでコンペアマッチで使用するジェネラルレジスタはGRB\_1です。

次にタイマZのクロックを決定しましょう。電源周波数が50Hzの場合を例に説明します。ゼロクロスとゼロクロスの間隔は電源周波数の半周期ですので(A)の時間は  $50\text{Hz}/2=10.0\text{msec}$  となります。また、制御を行う90%の範囲(B)は9.0msec、制御を行わない5%の範囲(C)は0.5msecとなります。タイマZは(A)の時間、つまり10.0msecはカウントしないとならないのでカウンタのクロックは余裕をみてCLK/8を選択することにします。CLK/8=400nsecで1カウント、タイマZは2バイトカウンタなのでこれなら最大26.2msecまでカウントすることが出来ます。範囲(B)を256段階で制御するので1変化量は時間で  $9.0\text{msec}/256=35.16\mu\text{sec}$ 、カウント値ですと  $9.0\text{msec}/256/400\text{nsec}=87.89$ 、約88カウントとなります。ジェネラルレジスタにセットする値は制御を行わない5%のカウント値+{出力n(1~254)×88カウント}となります。例えば出力50%の場合、(B)の半分D'11250と制御しない5%(C)のD'1250を加算した数D'12500をジェネラルレジスタにセットします。それぞれの時間とカウント値を表4-1にまとめます。

周波数	(A)	(B)	(C)	(B)/256
50Hz	10.0msec (D'25000)	9.0msec (D'22500)	0.5msec (D'1250)	35.16 $\mu$ sec (D'88)
60Hz	8.333msec (D'20833)	7.500msec (D'18750)	0.417msec (D'1042)	29.30 $\mu$ sec (D'73)



<表 5-1 各時間のカウント値>

ゼロクロス検出には“インプットキャプチャ”機能を使用します。これは外部からのエッジ入力でもカウント値をジェネラルレジスタにセットする機能で、この時タイマステータスレジスタにインプットキャプチャを知らせるフラグが立ちます。このフラグを利用してゼロクロスを検出します。ゼロクロス信号はFTIOC1に入力されているのでタイマステータスレジスタ1(TSR\_1)のビット2をチェックします。尚、フラグの検出はポーリングで行います。3章で検出信号とゼロクロスとの間にずれがあると説明しました。このずれは50Hzの場合約230  $\mu$  secあるので、ゼロクロスを検出し各処理を行った後、実際のゼロクロスまでウェイトをおきます。

このインプットキャプチャ機能はAC100Vの周波数を判定するのにも使用できます。ジェネラルレジスタGRC\_1にセットされたカウント値を加算して平均値を取り、その値で50Hzか60Hzかを判定すればよいのです。カウント値は50Hzで約D'25000、60Hzで約D'20833ですので、その中間D'22916より大きい小さいかで周波数を判定します。



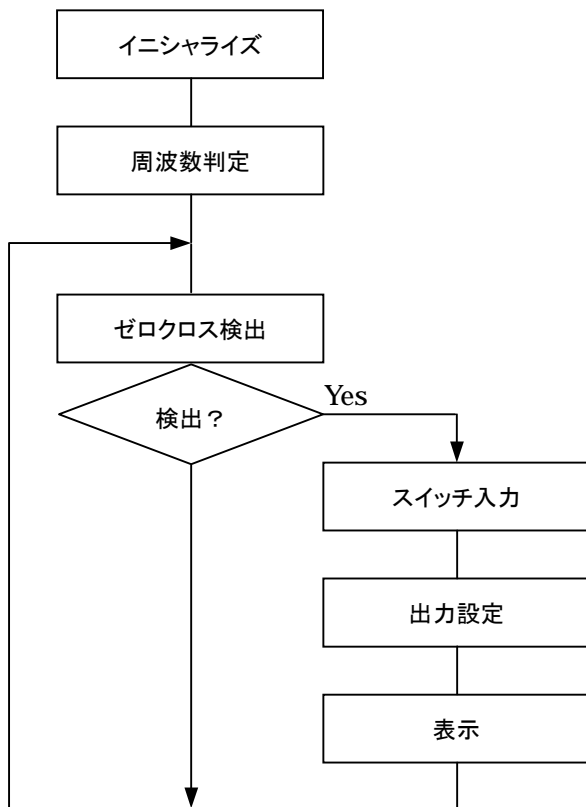
## ②操作スイッチ部

出力をコントロールするプッシュスイッチは単純にポートと接続されているだけなのでダブルリードしチャタリングを除去します。ダブルリードをかける時間はソフトで作っても良いのですがゼロクロスの信号があるのでそれを利用することにしましょう。周波数が 50Hz の地域なら 10msec、60Hz なら約 8.3msec のウェイトをかけられます。つまりゼロクロスを検出したらスイッチのリードも行うのです。尚、スイッチを押し続けた場合、ゼロクロス信号毎つまり AC の半周期毎に出力は変化します。

## ③表示部

出力レベルのモニタ表示は、出力の値を等間隔に分割してポート 5 の LED に表示します。ポート 5 を出力に設定し、点灯なら 1 を、消灯なら 0 を書きます。出力は無出力から最高出力まで 256 段階ですが、LED 全消灯は無出力、全点灯は最高出力なので表示 9 段階中 2 段階は予約済みです。ですので出力  $256 - 2$  (無出力と最高出力) = 254 を表示  $9 - 2$  (全消灯と全点灯) = 7 で割った数、 $254 / 7 = 36$  が表示を変化させる出力量となります。

概略フローを以下に示します。



### ・イニシャライズ

タイマ Z とポート 5 の初期化、バッファやフラグのクリアを行います。

### ・周波数判定

インプットキャプチャ機能を使い、AC100V の周波数を決定します。

### ・ゼロクロス検出

タイマステータスレジスタ 1 のビット 2 をチェックしてゼロクロスを検出したか判定します。

### ・スイッチ入力

スイッチが押されているかを判定し、押されていれば出力を変化させます。

### ・出力設定

出力値からタイマ Z を再セットし出力を変化させます。

### ・表示

出力値から LED の点灯数を求め表示します。

以上を踏まえて作成したプログラムが“ACPC\_1”です。次ページにリストを示します。

```

;-----
;
; FILE      :ACPC_1.src
; DATE      :Fri, Sep 26, 2003
; DESCRIPTION :Main Program
; CPU TYPE  :H8/3687
;
; This file is generated by Hitachi Project Generator (Ver.2.1)
;-----
; A Cパワーコントローラ
; タイマZのPWMモードを使用してACを位相制御する。ゼロクロ
; ス検出はインプットキャプチャで行い、TCNTをクリアする。
;-----
; .include "io3687F_equ.inc"
; .export  _main
;-----
;=====
;      メインプログラム
;=====
;----- インシャイス -----
_main:
    bsr     PIO_INIT:16      ;PIO インシャイス
    bsr     PWM_INIT:16     ;TimerZ インシャイス

    xor.w   r0,r0           ;フラグ・バッファ クリア
    mov.b   r0I,@Level
    mov.b   r0I,@USw_STA
    mov.b   r0I,@DSw_STA

    bsr     Check_Hz:16

;----- メインループ -----
_main_loop:
    mov.b   @TSR_1,r0I      ;タイマステータスレジスタ ch1
    btst   #2,r0I          ;ゼロクロス検出?
    bne    _main_zeroX     ;NE=ゼロクロス検出
    bra    _main_loop

;----- ゼロクロス検出 -----
_main_zeroX:
;----- タイマステータスレジスタ -----
    bclr   #2,r0I          ;インプットキャプチャ C
    mov.b   r0I,@TSR_1     ;クリア

;----- タイマアウトプットコントロールレジスタ -----
    mov.b   #B'00100000,r0I ;出力初期化
    mov.b   r0I,@TOCR      ;FT10B1=Hi / other=Low

;----- タイマスタートレジスタ -----
    mov.b   #B'11111100,r0I ;TimerZ
    mov.b   r0I,@TSTR      ;動作停止

    mov.b   @PDR6,r0I      ;スイッチ リード
    btst   #0,r0I          ;Up_Sw チェック
    beq    _UpSw
    btst   #1,r0I          ;Down_Sw チェック
    beq    _DownSw
    xor.b   r0I,r0I        ;スイッチ入力無し
    mov.b   r0I,@USw_STA   ;スイッチフラグ クリア
    mov.b   r0I,@DSw_STA
    bra    _zeroX_00

;----- Up スイッチ -----
_UpSw:
    bsr     Up_Switch      ;出力+1
    xor.b   r0I,r0I
    mov.b   r0I,@DSw_STA
    bra    _zeroX_00

;----- Down スイッチ -----
_DownSw:
    bsr     Down_Switch   ;出力-1
    xor.b   r0I,r0I
    mov.b   r0I,@USw_STA
    bra    _zeroX_00

;----- 表示と出力の決定 -----
_zeroX_00:
    bsr     Level_DISP:16 ;出力の表示

    bsr     Set_GRB1:16   ;GRB1 のセット

    xor.l   er6,er6
    add.w   #H'02FB,r6    ;H'02FB=230usec
    bsr     Wait:16

    bsr     Start_TZ:16  ;タイマZの出力とスタート

    bra    _main_loop:16

;----- サブルーチンプログラム -----
;----- Up スイッチ : 出力 + 1 -----
Up_Switch:
    mov.b   @USw_STA,r0I  ;ステータス チェック
    beq    ?000           ;ステータス=0 ?
    cmp.b   #1,r0I        ;ステータス=1 ?
    beq    ?100
    xor.b   r0I,r0I      ;その他ステータスは無し
    mov.b   r0I,@USw_STA ;ステータス クリア
    rts

;----- ステータス=0 : ウェイト -----
?000:
    mov.b   #1,r0I        ;ダブルリードの為 ウェイト
    mov.b   r0I,@USw_STA ;次のステータスをセット
    rts

;----- ステータス=1 : Level+1 -----
?100:
    mov.b   @Level,r0I    ;スイッチ押され確認
    cmp.b   #H'FF,r0I     ;出力バッファ リード
    beq    ?102           ;Level=H'FF : 最高出力
    inc.b   r0I           ;最高出力なら+1 しない
    mov.b   r0I,@Level    ;出力+1

?102:
    rts

```

【ADPC\_1.src (1/3)】

Downスイッチ：出力 - 1	周波数確認と定数のセット
<pre> Down_Switch: mov.b    @DSw_STA, r0l    ;ステータス チェック beq      ?000             ;ステータス=0 ? cmp.b    #1, r0l          ;ステータス=1 ? beq      ?100 xor.b    r0l, r0l         ;その他ステータスは無し mov.b    r0l, @DSw_STA    ;ステータス クリア rts  ;----- ステータス=0 : ウェイト ----- ?000: mov.b    #1, r0l          ;ダブリードの為 ウェイト mov.b    r0l, @DSw_STA    ;次のステータスをセット rts  ;----- ステータス=1 : Level+1 ----- ?100: mov.b    @Level, r0l     ;スイッチ押され確認 bne      ?102            ;出力バッファ リード mov.w    #H'FFFF, r0     ;Level=H'00 : 無出力 bra      ?104            ;無出力なら-1 しない  ?102: dec.b    r0l              ;出力-1 mov.b    r0l, @Level  ?104: rts  ;----- ジェネラルレジスタ B 1 のセット ----- Set_GRB1: mov.b    @Level, r0l beq      ?000             ;無出力 ? cmp.b    #H'FF, r0l      ;最高出力 ? beq      ?100  ;----- H'01 ~ H'FE ----- neg.b    r0l              ;0-R0L mov.b    @CNST_1per, r1l mulxu.b r1l, r0           ;R0*R1L=R0 mov.w    @CNST_5per, r1 add.w    r1, r0 bra      ?200  ;----- H'00 ----- ?000: mov.w    #H'0001, r0 bra      ?200  ;----- H'FF ----- ?100: mov.w    #H'FFFF, r0 bra      ?200  ;----- GRB1 へセット ----- ?200: mov.w    r0, @GRB_1 rts </pre>	<pre> Check_Hz: mov.b    #B'11111110, r0l ;TimerZ ch1 mov.b    r0l, @TSTR       ;動作開始  ?000: ;タイマステータスレジスタ_1 mov.b    @TSR_1, r0l     ;インพุットキャプチャ_C チェック btst     #2, r0l         ;セロクロス検出? beq      ?000            ;NE=セロクロス検出 bclr     #2, r0l         ;インพุットキャプチャ_C mov.b    r0l, @TSR_1     ;クリア  mov.w    @GRC_1, r0 mov.w    r0, @H'F800  xor.l    er1, er1        ;ER1=加算バッファ mov.b    #D'16, r2l      ;R2L=加算回数  ?002: ;タイマステータスレジスタ_1 mov.b    @TSR_1, r0l     ;インพุットキャプチャ_C チェック btst     #2, r0l         ;セロクロス検出? beq      ?002            ;NE=セロクロス検出 bclr     #2, r0l         ;インพุットキャプチャ_C mov.b    r0l, @TSR_1     ;クリア mov.w    @GRC_1, r0     ;GRC_1=セロクロス間のカウント値 xor.w    e0, e0 add.l    er0, er1        ;加算 dec.b    r2l             ;回数-1 bne      ?002  mov.l    er1, @H'F810  shl.r.l  er1 shl.r.l  er1 shl.r.l  er1 shl.r.l  er1  mov.w    r1, @H'F820  cmp.w    #H'5985, r1 bcs      ?60  ?50: mov.b    #D'88, r0l mov.w    #D'1250, r1 bra      ?100  ?60: mov.b    #D'73, r0l mov.w    #D'1042, r1 bra      ?100  ; ?100: mov.b    r0l, @CNST_1per mov.w    r1, @CNST_5per ;タイマステータスレジスタ mov.b    #B'11111100, r0l ;TimerZ mov.b    r0l, @TSTR       ;動作停止  rts </pre>

【ADPC\_1.src (2/3)】

```

;-----
;
; タイマZの出力とスタート
;-----
Start_TZ:
;タイマカウンタ
xor.w    r0,r0          ;タイマカウンタ ch1
mov.w    r0,@TCNT_1    ;クリア

mov.b    @Level,r01     ;出力チェック
beq      ?000           ;無出力?
cmp.b    #H'FF,r01     ;最大出力?
beq      ?100

;----- Level = H'01 ~ H'FE -----
;タイマアウトプットコントロールレジスタ
mov.b    #B'00100000,r01 ;出力初期化
mov.b    r01,@TOCR      ;FT10B1=Hi / other=Low

;タイマスタートレジスタ
mov.b    #B'11111110,r01 ;TimerZ ch1
mov.b    r01,@TSTR     ;動作開始
rts

;----- Level = H'00 -----
?000:
;タイマアウトプットコントロールレジスタ
mov.b    #B'00100000,r01 ;出力初期化
mov.b    r01,@TOCR      ;FT10B1=Hi / other=Low
rts

;----- Level = H'FF -----
?100:
;タイマアウトプットコントロールレジスタ
mov.b    #B'00000000,r01 ;出力初期化
mov.b    r01,@TOCR      ;FT10B1&other=Low
rts

;-----
;
; P5LEDへ出力表示
;-----
; :点灯 / :消灯
;無出力 (Level=H'00) :
;制御点灯(Level=H'01~H'FE) :
;
;最高出力(Level=H'FF) :
Level_DISP:
xor.b    r11,r11       ;R1L=表示データ
mov.b    @Level,r01    ;出力リード
beq      ?100         ;無出力?
cmp.b    #H'FF,r01     ;最高出力?
bne      ?000
mov.b    #H'FF,r11     ;最高出力=全点灯
bra      ?100

?000:
mov.b    #D'36,r0h     ;D'254/7 = D'36
?002:
;表示データ作成
bset     #7,r11
rotl.b   r11
sub.b    r0h,r01
bcs     ?004
bra      ?002

?004:
bset     #0,r11       ;H'01~H'1Fは点灯
?100:
mov.b    r11,@PDR5    ;表示
rts
;-----
;
; P I Oイニシャライズ
;-----
PIO_INIT:
mov.b    #B'11111111,r01 ;P5:表示用
mov.b    r01,@PCR5
mov.b    #B'00000000,r01 ;P60,61:スイッチ入力
mov.b    r01,@PCR6
rts

;-----
;
; PWM(タイマZ)イニシャライズ
;-----
PWM_INIT:
;タイマコントロールレジスタ
mov.b    #B'10100011,r01 ;GRCインプットキャプチャクリア
mov.b    r01,@TCR_1     ;CLK/8

;タイマI/Oコントロールレジスタ
mov.b    #B'10011000,r01 ;GRA:-
mov.b    r01,@TIORA_1   ;GRB:アウトプットコンパリア出力
mov.b    #B'10001100,r01 ;GRC:インプットキャプチャHi_edge
mov.b    r01,@TIORC_1  ;GRD:-

;タイマアウトプットコントロールレジスタ
mov.b    #B'00100000,r01 ;FT10B1=Hi / other=Low
mov.b    r01,@TOCR

;シフトレジスタ
mov.w    #H'FFFF,r0
mov.w    r0,@GRB_1

;タイマアウトプットインポートレジスタ
mov.b    #B'10011111,r01 ;FT10A,B許可
mov.b    r01,@TOER

rts

;-----
;
; ウェイト
;-----
;Loop_count = (Wait_Time/300n)-(22/6)
Wait:
dec.l    #1,er6        ;2
bne      Wait          ;4
rts      ;8

;-----
;
; ワークエリア
;-----
.section D,data
CNST_1per .res.b 1 ;AC1%のカウント値
.align 2
CNST_5per .res.w 1 ;AC5%のカウント値
Level .res.b 1 ;出力レベル
USw_STA .res.b 1 ;Upスイッチステータス
DSw_STA .res.b 1 ;Downスイッチステータス
.align 2

;-----
;
; .end
;-----

```

【ADPC\_1.src (3/3)】

## 6 ハードの製作と動作チェック

### ■ハードの製作

ユニバーサル基板にハードを組み上げます。プリント基板と違いユニバーサル基板は全ての配線を自分で結線しなければなりません。けれども回路図を見ながら部品をハンダ付けしていく事によってハードの構成をより理解することと思います。まず、工具・部品の確認を行いましょう。下記の部品表と照らし合わせて全ての部品が揃っている事を確認して下さい。

#### ・工具

ハンダごて、ハンダ、ニッパ、ワイヤストリッパ、ピンセット、+ドライバ、テスタ

#### ・部品

TK-3687オプション ACパワーコントローラ 部品表	全数:	1
------------------------------	-----	---

部品番号	型名, 規格	メーカー	数量	全数	備考
1	■ユニバーサル基板				
2	U1	TLP521-2	東芝	1	1
3	SSR1	G3M-102PL/DC5V	オムロン	1	1
4	BA1	TNR181K005B		1	1
5	R1	33K $\Omega$ , 2W		1	1
6	R2	10k $\Omega$		1	1
7	SW1,2	SKHHAK/AM/DC	ALPS	2	2
8	CN1	TS-125P/4極	キムデン	1	1
9	CN2	10pin 丸ピンソケット		2	2 20pinを2つに切断 ※1
10	ラッピングケーブル	50cm		1	1 結線用
11	メッキ線				AC, Vcc, GND等半田面結線用 ※2
12	ゴム足			4	4
13	接続ケーブル	A10-N100-A	Hitaltech	1	1
14	ACコード			1	1 切断して使用
15	基板	B6082	東洋リンクス	1	1
16					

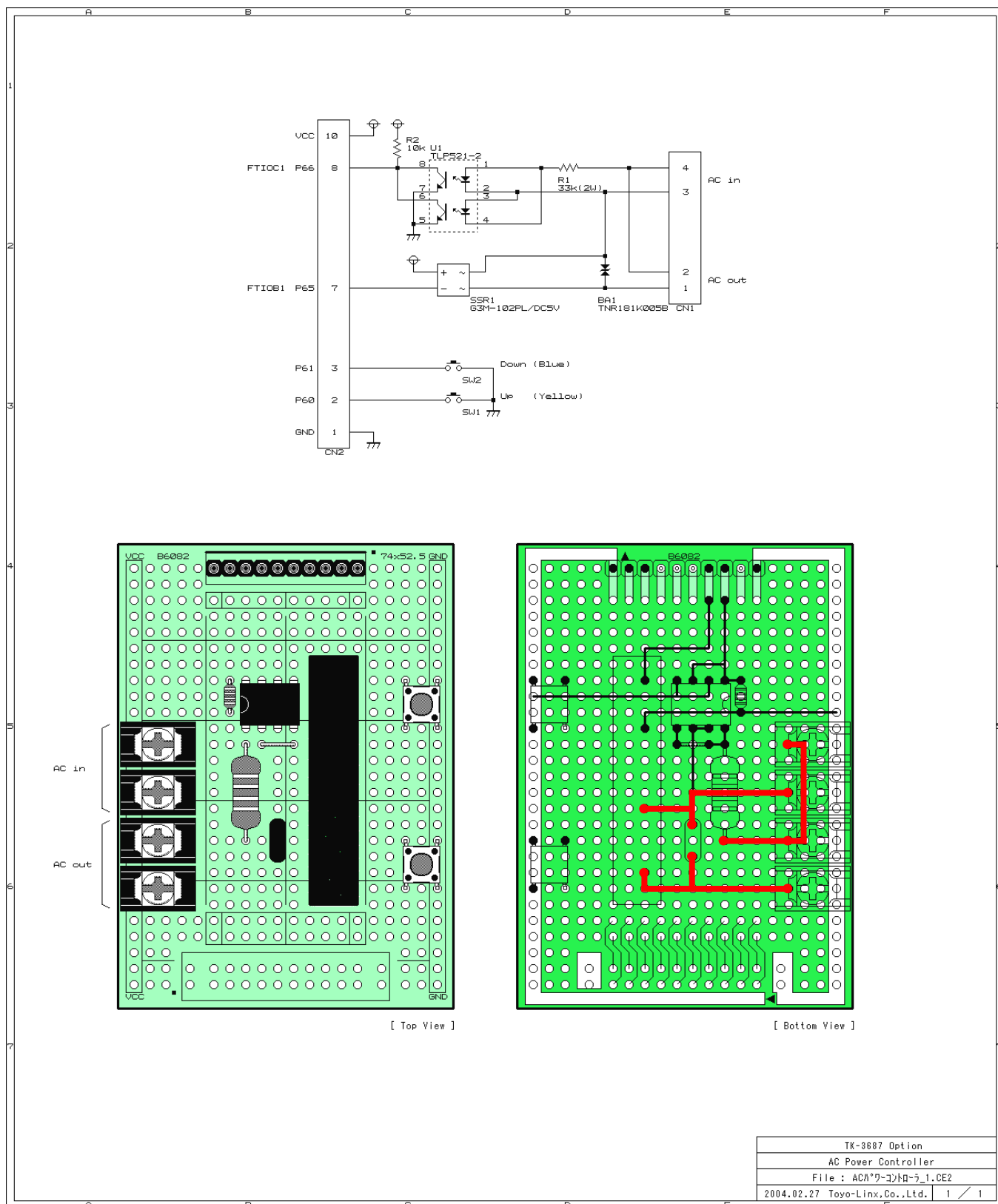
※相当品を使用する場合があります。

※1 1つはTK-3687・CN6に実装します。

※2 ラッピングケーブルの被覆を剥し2本をよじって使用します。また抵抗やコンデンサのリードも流用できます。

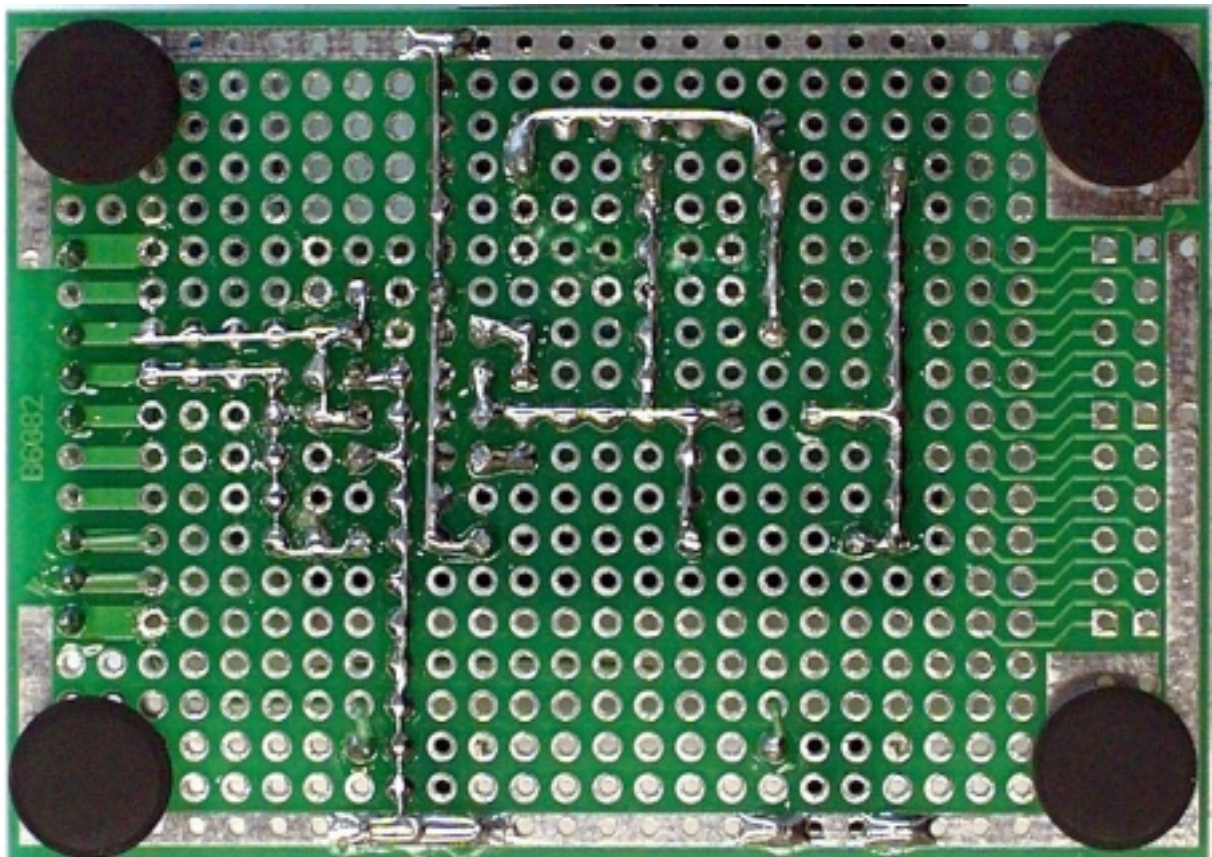
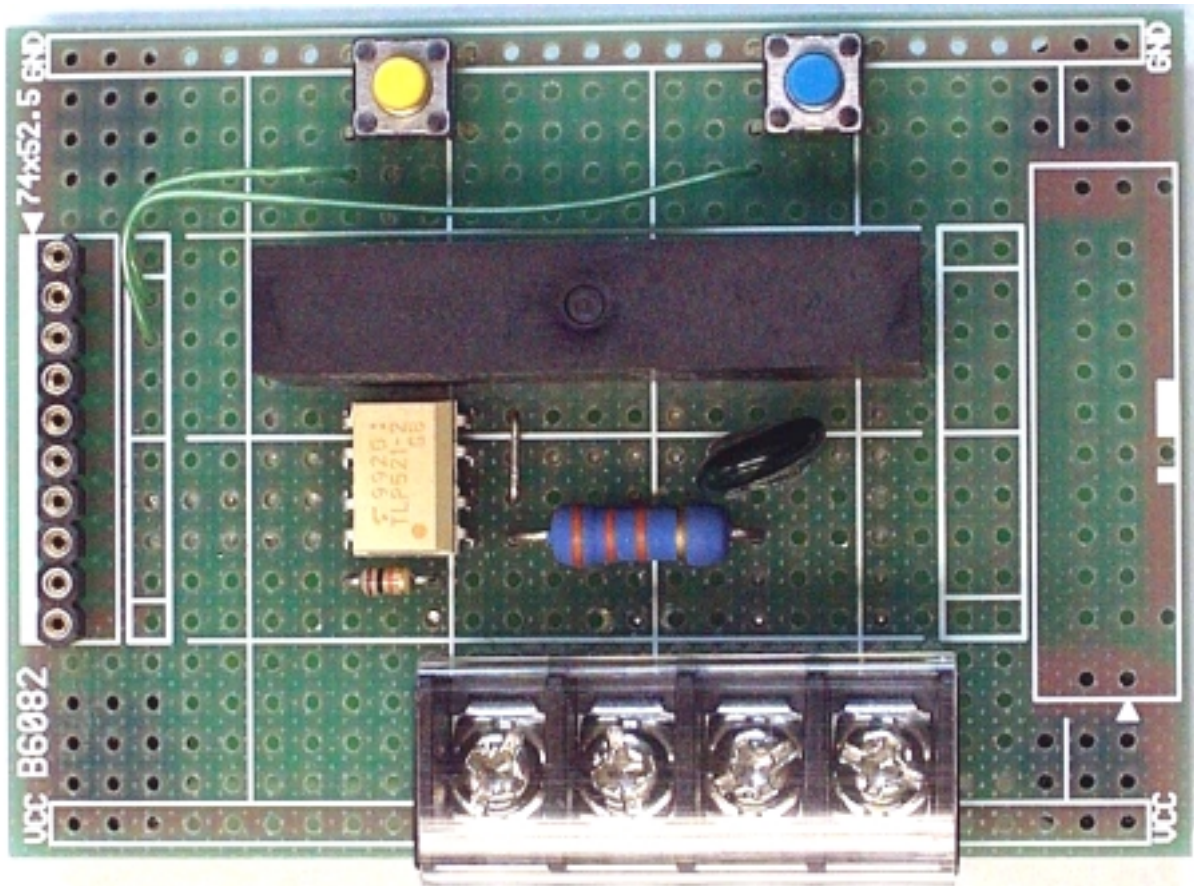
部品の確認が済みましたら、いよいよ組み立てです。13~15 ページの回路図・実装図、完成写真等をよく見ながらハンダ付けを行って下さい。ハンダ付けによるケーブル配線は大変なので、電源や GND、交差しない信号線等はハンダ面でメッキ線や部品の余ったリード線を流用して接続します(P.14 完成写真下段を参照)。ハンダ面のみでは配線しきれない信号線はラッピングケーブルで配線していきます。尚、ラッピングケーブルでの配線が初めての方は15 ページの結線方法を参照して下さい。

# 回路図・実装図



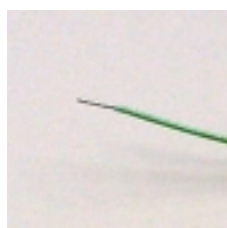


完成写真

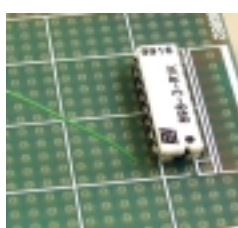


## ■ラッピングケーブルでの結線方法

ハンダ面で結線されない配線はラッピングケーブルで結線していきます。ここではラッピングケーブルでの結線の仕方を説明します。



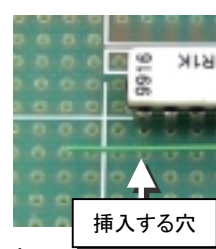
1. 被覆を剥きます (5mm 位)



2. 部品面から穴に通します



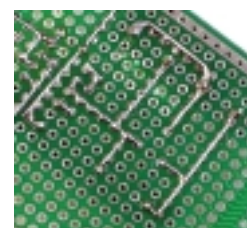
3. ハンダ付けする部品の足にラッピングケーブルを絡げてからハンダ付けします



4. 結線先までケーブルを這わせ、挿入する穴から3つ先の穴辺りでカットします  
1~3の手順でハンダ付けします

## ■AC100V 信号線を太く結線する

AC100V がかかるライン、特に13 ページ実装図の赤で示している結線は、太いリード線やメッキ線を二重に配線しハンダで埋めるなど、なるべく太く結線します。



## ■AC コードの接続

AC コードは真中で切断し、被覆を剥いて芯線を端子台のネジで挟んで接続します。AC-in 側にプラグ(オス)、AC-out 側にソケット(メス)を接続して下さい。



## ■配線の確認

全ての配線を終わったら配線の確認を行います。テストで回路図通り結線されているか導通チェックを行って下さい。又、電源とグラウンド、AC100V ラインがショートしていないか、併せて念入りに確認します。

## ■動作チェック

配線の確認を終えたら、動作チェックを行います。付属の CD-ROM 内にプログラムの HEX ファイルが収録されていますのでそれを使用します。まず組み立てたユニバーサル基板と CPU 基板とをケーブルで接続し、AC-out に白熱灯を、AC-in をコンセントへと接続します。次にハイパーH8 の LG コマンドで CD-ROM 内のプログラムをロードして実行します。プログラムは、CD-ROM:¥TK-3687¥オプション¥AC パワーコントローラ¥プログラム ¥ACPC\_1.mot です。基板上の Up スイッチを押すと徐々に白熱灯が明るくなり、Down スイッチを押すと徐々に暗くなれば正常に動作しています。尚、このキットで使用している SSR は 200W まで接続できます。

もし動作しない場合や基板から焼けるような臭いや煙が出るなど異変に気づいた時は、すぐに電源を落とし、各配線をもう一度確認して下さい。この時確実に電源を落としたのを確認してから基板に触れてください。感電する恐れがあります。また、焼けるような臭いや煙などが出た場合は部品が発熱している場合がありますので注意してください。



## 7 付録

プログラム中で使用したタイマZの内蔵レジスタと各ビットの意味、設定値を記します。尚、更に詳しく知りたい方は“H83687 シリーズ ハードウェアマニュアル”を参照して下さい。

### ・タイマコントロールレジスタ【TCR\_1】

カウンタクロック選択、カウンタクリア要因の選択を行います。各チャンネル毎に1本、計2本のTCRがあります。

7	6	5	4	3	2	1	0
CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
カウンタクリア			クロックエッジ		タイマプリスケアラ		
1	0	1	0	0	0	1	1
GRC インプットキャプチャで TCNT クリア			外部クロック選択時に使用		内部クロック、1/8 でカウント		

### ・タイマ I/O コントロールレジスタ A\_1【TIORA\_1】

GRA、GRB の機能を設定します。

7	6	5	4	3	2	1	0
-	IOB2	IOB1	IOB0	-	IOA2	IOA1	IOA0
-	IO コントロール B			-	IO コントロール A		
1	0	0	1	1	0	0	0
-	GRB のコンペアマッチで 0 出力			-	コンペアマッチによる端子出力禁止		

### ・タイマ I/O コントロールレジスタ C\_1【TIORC\_1】

GRC、GRD の機能を設定します。

7	6	5	4	3	2	1	0
-	IOD2	IOD1	IOD0	-	IOC2	IOC1	IOC0
-	IO コントロール D			-	IO コントロール C		
1	0	0	0	1	1	0	0
-	コンペアマッチによる端子出力禁止			-	立ち上がりエッジで GRC ヘインプットキャプチャ		

### ・タイマアウトプットコントロールレジスタ【TOCR】

コンペアマッチが起こるまでの初期出力を設定をします。

7	6	5	4	3	2	1	0
TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0
出力レベルセレクトチャンネル 1				出力レベルセレクトチャンネル 0			
0	0	1	0	0	0	0	0
FTIOB1、1 出力							

・ジェネラルレジスタ A、B、C、D【GRA、GRB、GRC、GRD】

GR は 16 ビットのレジスタで各チャンネルに 4 本、計 8 本あります。

GRB には出力する比率に相当する値をセットします。ちなみに最高出力、無出力時はカウンタを止めてレベル出力としますが、仮に動作してしまっても出力に影響ないように最高出力時には H'0001 を、無出力時は H'FFFF をセットします。

GRC は周波数判定時に使用します。ゼロクロスを検出後に GRC の値を読み取ることによってゼロクロス間のカウントを知ることが出来ます。

・タイマアウトプットマスタイネーブルレジスタ【TOER】

チャンネル 0、1 の出力を許可／禁止します。

7	6	5	4	3	2	1	0
ED1	EC1	EB1	EA1	ED0	EC0	EB0	EA0
マスタイネーブルチャンネル 1				マスタイネーブルチャンネル 0			
1	0	0	1	1	1	1	1
使用する FTIOB1, FTIOC1 の出力を許可							

・タイマスタートレジスタ【TSTR】

TCNT の動作／停止を選択します。

7	6	5	4	3	2	1	0
-	-	-	-	-	-	STR1	STR0
-	-	-	-	-	-	カウンタスタート	
1	1	1	1	1	1	1	0
チャンネル 1 動作							

・タイマステータスレジスタ【TSR】

タイマのステータスを表します。各チャンネル毎に 1 本、計 2 本の TCR があります。

7	6	5	4	3	2	1	0
-	-	UDF	OVF	IMFD	IMFC	IMFB	IMFA
-	-	アンダー フロー フラグ	オーバー フロー フラグ	インプット キャプチャ コンペアマッチ フラグ D	インプット キャプチャ コンペアマッチ フラグ C	インプット キャプチャ コンペアマッチ フラグ B	インプット キャプチャ コンペアマッチ フラグ A
IMFC=1 でゼロクロス検出、検出後はクリアすること							

※ステータスなので設定値はありません。

★ご質問、お問い合わせはメール又は FAX で、、、

## (株)東洋リンクス

〒102-0093 東京都千代田区平河町 1-2-2 朝日ビル

TEL: 03-3234-0559 / FAX: 03-3234-0549

URL: <http://www2.u-netsurf.ne.jp/~toyolinx>

E-Mail: [toyolinx@va.u-netsurf.jp](mailto:toyolinx@va.u-netsurf.jp)

※本書の内容は将来予告無しに変更することがあります(2007年1月作成)

20070125