

```

*****
;
;
; * Title      : 1Chip Transmitter
; * Version    :
; * Last updated : SEP.20.2005
; * Target     : AT90S1200
;
; * DESCRIPTION
; *           最終バージョン
; *           周波数調整済み
;
; *           電鍵モードは電鍵(トット)ONのままPOWER ON
; *
;
;
;
; *****

```

```

.include "1200def.inc"

```

```

*****
;
; * 定数,PORT *
; *****
;
;

```

```

.equ LD4 = 2 ; PB2 LED DATA 4
.equ LD5 = 3 ; PB3 LED DATA 5 H:LED1-5 L:LED6-11
.equ CAR = 4 ; PB4 キャリア出力
.equ RY = 5 ; PB5 ANTIL- HでON(TX)
.equ SWDOT = 6 ; PB6 DOT 入力
.equ SWDAS = 7 ; PB7 DASH 入力

```

```

.equ BZ = 0 ; PD0 SIDE TONE プザ-
.equ LD0 = 1 ; PD1 LED DATA 0
.equ LD1 = 2 ; PD2 LED DATA 1
.equ LD2 = 3 ; PD3 LED DATA 2
.equ LD3 = 5 ; PD5 LED DATA 3
.equ CR = 6 ; PD6 CR充電出力

```

```

.equ OUTH = 0xD0 ; キャリアH出力 PORTB用1ハイトデータ
.equ OUTL = 0xC0 ; キャリアL出力 PORTB用1ハイトデータ

```

```

.equ K1 = 20 ; CRチャージ時間生成カウンタ 下位
.equ K2 = 20 ; CRチャージ時間生成カウンタ 上位
.equ K3 = 226 ; TMRO定数(BZ周波数)
.equ GTL = 111 ; ゲ-ト時間生成カウンタ 下位
.equ GTH = 122 ; ゲ-ト時間生成カウンタ 上位
.equ FQ = 69 ; カウント値=>周波数 変換ワセツ値

```

```

*****
;
; * レジスタ割付,フラグ *
; *****
;
;

```

```

.def out_H = r16 ; キャリアH出力 PORTB用1ハイトデータ保管場所
.def out_L = r17 ; キャリアL出力 PORTB用1ハイトデータ保管場所
.def tmr_K = r18 ; TMRO定数保管場所
.def temp = r19 ; 汎用カウンタ
.def cnt1 = r20 ; カンタ1 充電時間LSB ゲ-ト時間生成
.def cnt2 = r21 ; カンタ2 充電時間MSB ゲ-ト時間生成
.def cnt3 = r22 ; カンタ3 トット出力回数

```

```

.def bzflug = r23 ; 現在のプザ-状態フラグ 0x00:プザ-OFF 0x01:プザ-ON
.def memflug = r24 ; トット/タツシュメモリフラグ(長短点メモリ)
; 0x01:トット/タツシュ中にトットON 0x02:トット/タツシュ中にタツシュON
; 0x04:ス^ -ス中にトットON 0x08:ス^ -ス中にタツシュON
.def dotflug = r25 ; 現在のトット/タツシュ処理 0x00:タツシュ処理中 0x01:トット処理中

```

```

*****
;
; * リセット、割り込みベクタ設定
; *****
;
;

```

```

.CSEG
;

```

```

    rjmp  RESET                ; Reset   初期化へ
    reti                          ; INTO   割り込み使わない
    reti                          ; TMRO   割り込み使わない
    reti                          ; コンパレータ 割り込み使わない

;*****
;* RESET ここからスタート
;*****
RESET:
    ldi    temp,0x3C           ; PORTB=ii0000ii
    out    DDRB,temp          ;
    ldi    temp,0xC0          ; PB6,7 フォルアップ
    out    PORTB,temp         ;

    ldi    temp,0xEF          ; PORTD=000i0000
    out    DDRD,temp          ;
    ldi    temp,0x3E          ; PD4 フォルアップ
    out    PORTD,temp         ;

    ldi    temp,0x00          ;
    out    ACSR,temp          ; コンパレータON,割り込み禁止
    ldi    temp,0x00          ;
    out    SREG,temp          ; 全割り込み禁止
    ldi    temp,0x00          ;
    out    GIMSK,temp         ; INTO割り込み禁止
    ldi    temp,0x00          ;
    out    TIMSK,temp         ; TMRO割り込み禁止

    ldi    out_H,OUTH         ; キャリアH出力定数をRdに設定
    ldi    out_L,OUTL         ; キャリアL出力定数をRdに設定

    rcall  WAIT1              ; 50msec ウェイト

    sbis   PINB,SWDOT         ; 電鍵モードチェック ドット(電鍵) ON?
    rjmp   DENSTART          ; Yes! 電鍵モードへ

    rcall  LEDCHK             ; 11個LED順次点灯

;*****
;* エレキーモード メインルーチン
;* DOT/DASH 入力待ち この間(受信中のみ)周波数表示をする。
;*****
MAIN:

    sbis   PINB,SWDOT         ; ドット ON?
    rjmp   DOT                ; Yes! ドット出力へ

    sbis   PINB,SWDAS        ; ダッシュ ON?
    rjmp   DASH               ; Yes! ダッシュ出力へ

    rcall  FDISP              ; 周波数表示

    andi   out_H,0xDF         ; キャリアH出力定数 RY OFF
    andi   out_L,0xDF         ; キャリアL出力定数 RY OFF
    out    PORTB,out_L        ; 出力

    rjmp   MAIN               ; 繰り返す

;*****
;* ドット出力ルーチン
;*****
DOT:

    ldi    cnt3,1             ; CNT3=1 ドット1個ぶん
    ldi    dot_flg,1         ; 現在ドット処理中 dot_flg=1
    rjmp   CRO                ; CR充電へ

;*****
;* ダッシュ出力ルーチン
;*****
DASH:

    ldi    cnt3,3             ; CNT3=3 ダッシュ=ドット3個ぶん
    clr    dot_flg           ; 現在ダッシュ処理中 dot_flg=0
    rjmp   CRO                ; CR充電へ

```

```

;*****
;* ドット/ダッシュ送信 CR充電ルーチン 充電はドット時1回実行 ダッシュ時3回実行
;*****

```

```

CR0:
    ldi    tmr_K,K3          ;TMRO定数をRdに設定
    out    TCNT0,tmr_K      ;TMRO定数設定
    ldi    temp,3           ;
    out    TCCR0,temp       ;TMROスタート Clock/64
    ldi    temp,2           ;
    out    TIFR,temp        ;TMRO TOVOクリア

    sbi    PORTD,BZ         ;ブザーON
    sbr    bzflug,1         ;ブザーフラグ 1に設定
    clr    memflug          ;長短メモリフラグクリア

    ori    out_H,0x20       ;キャリアH出力定数 RY ON
    ori    out_L,0x20       ;キャリアL出力定数 RY ON

```

```

;充電ルーチン -----

```

```

CR01:
    out    PORTB,out_H      ;      OUT=H
    ldi    cnt1,K1          ;CNT1=K1          充電時間設定L
    ldi    cnt2,K2          ;CNT2=K2          充電時間設定H
    out    PORTB,out_L      ;      OUT=L
    nop
    nop

    out    PORTB,out_H      ;      OUT=H          充電開始
    sbi    PORTD,CR         ;充電開始
    out    PORTB,out_L      ;      OUT=L
    sbi    DDRD,CR          ;充電端子OUTに変更

CR02:
    out    PORTB,out_H      ;      OUT=H          充電完了待ちループ
    nop
    dec    cnt1             ;CNT1 マイナス1
    out    PORTB,out_L      ;      OUT=L
    brne   CR02             ;CNT1が0でない CR02へ
    ldi    cnt1,K1          ;CNT1=K1          充電時間再設定L

```

```

;ブザー用タイマーオーバーフローチェック -----

```

```

    out    PORTB,out_H      ;      OUT=H          ブザー用TMROチェック
    in     temp,TIFR        ;TIFR読み込み
    andi   temp,0x02        ;TOV0=H?(オーバーフロー?)
    out    PORTB,out_L      ;      OUT=L
    brne   CR20             ;Yes! CR20へ          オーバーフローした
    nop
    ;No!

CR03:
    out    PORTB,out_H      ;      OUT=H          充電完了待ちループ
    nop
    dec    cnt2             ;CNT2 マイナス1
    out    PORTB,out_L      ;      OUT=L
    brne   CR02             ;CNT2が0でない CR02へ          完了まで繰り返す
    nop

```

```

CR04:

```

```

    out    PORTB,out_H      ;      OUT=H          充電完了
    cbi    DDRD,CR         ;CR INに戻す
    out    PORTB,out_L      ;      OUT=L
    cbi    PORTD,CR        ;CR 充電終了

    out    PORTB,out_H      ;      OUT=H
    nop
    nop
    out    PORTB,out_L      ;      OUT=L
    rjmp   DOTDASH         ;放電へ

```

```

;充電中のブザーON/OFF処理 -----

```

```

CR20:
    out    PORTB,out_H      ;      OUT=H          タイム再起動
    nop
    out    TCNT0,tmr_K      ;TMRO定数再設定
    out    PORTB,out_L      ;      OUT=L
    ldi    temp,2           ;
    out    TIFR,temp        ;TMRO TOVOクリア

    out    PORTB,out_H      ;      OUT=H
    nop
    andi   bzflug,1         ;ブザーフラグ=1?
    out    PORTB,out_L      ;      OUT=L

```

```

brne CR22 ;Yes! CR22へ
nop ;
CR21: out PORTB,out_H ; OUT=H 前回BZはOFFなのでONにする
sbr bzfl ug,1 ;ブザーフラグ 1に設定
nop ;
out PORTB,out_L ; OUT=L
sbi PORTD,BZ ;ブザーON
;
out PORTB,out_H ; OUT=H
nop ;
nop ;
out PORTB,out_L ; OUT=L
rjmp CR03 ;CR03へ戻る
CR22: out PORTB,out_H ; OUT=H 前回BZはONなのでOFFにする
cbr bzfl ug,1 ;ブザーフラグ 0に設定
nop ;
out PORTB,out_L ; OUT=L
cbi PORTD,BZ ;ブザーOFF
;
out PORTB,out_H ; OUT=H
nop ;
nop ;
out PORTB,out_L ; OUT=L
rjmp CR03 ;CR03へ戻る

```

```

;*****
;* ドットダッシュ送信 CR放電ルーチン 放電はドット時1回実行 ダッシュ時3回実行
;*****

```

DOTDASH:

```

;放電完了チェック -----
out PORTB,out_H ; OUT=H コンパレータチェック
in temp,ACSR ;ACSR読み込み
andi temp,0x20 ;ACO=L?
out PORTB,out_L ; OUT=L
breq D050 ;Yes! D050へ 放電完了
nop ;No!

```

;ドットダッシュメモリ処理 -----

```

ME1: out PORTB,out_H ; OUT=H ドットメモリ
in temp,PINB ;PORT読み込み
andi temp,0x40 ;ドットON?
out PORTB,out_L ; OUT=L
brne ME2 ;No! ME2へ
ori memfl ug,0x01 ;Yes! フラグ へ1をORする

```

```

ME2: out PORTB,out_H ; OUT=H ダッシュメモリ
in temp,PINB ;PORT読み込み
andi temp,0x80 ;ダッシュON?
out PORTB,out_L ; OUT=L
brne ME3 ;No! ME3へ
ori memfl ug,0x02 ;Yes! フラグ へ2をORする

```

;ブザー用タイマーオーバーフラグチェック -----

```

ME3: out PORTB,out_H ; OUT=H ブザー用TMR0チェック
in temp,TIFR ;TIFR読み込み
andi temp,0x02 ;TOV0=H?(オーバーフラグ?)
out PORTB,out_L ; OUT=L
breq DOTDASH ;No! DOTDASHへ繰り返す
nop ;Yes! オールフラグしていない

out PORTB,out_H ; OUT=H ブザーON/OFF処理へ
nop ;
nop ;
out PORTB,out_L ; OUT=L
rjmp D01 ;D01へ

```

;ダッシュは3回繰り返す -----

```

D050: out PORTB,out_H ; OUT=H 放電完了
nop ;
dec cnt3 ;CNT3 マイナス1 繰返し回数の判断
out PORTB,out_L ; OUT=L
breq SP1 ;スペースへ
1ドット/3ドット終了、スペースへ

```

```

D051:  nop      ;
      out     PORTB,out_H      ; OUT=H          タッチは3回繰り返す 長距離ジャンプ
      nop
      nop
      out     PORTB,out_L      ; OUT=L
      rjmp    CR01            ;CR01充電へ

```

;放電中のブザーON/OFF処理-----

```

D01:  out     PORTB,out_H      ; OUT=H          タイマ-再起動
      nop
      out     TCNT0,tmr_K     ; TMRO定数再設定
      out     PORTB,out_L     ; OUT=L
      ldi     temp,2
      out     TIFR,temp       ; TMRO TOVOクリア

      out     PORTB,out_H     ; OUT=H
      nop
      andi    bzflag,1        ; ブザーフラグ=1?
      out     PORTB,out_L     ; OUT=L
      brne    D03             ; Yes! D03へ
      nop

D02:  out     PORTB,out_H     ; OUT=H          前回BZはOFFなのでONにする
      sbr     bzflag,1        ; ブザーフラグ 1に設定
      nop
      out     PORTB,out_L     ; OUT=L
      sbi     PORTD,BZ        ; ブザーON

      out     PORTB,out_H     ; OUT=H
      nop
      nop
      out     PORTB,out_L     ; OUT=L
      rjmp    DOTDASH        ; DOTDASHへ戻る

D03:  out     PORTB,out_H     ; OUT=H          前回BZはONなのでOFFにする
      cbr     bzflag,1        ; ブザーフラグ 0に設定
      nop
      out     PORTB,out_L     ; OUT=L
      cbi     PORTD,BZ        ; ブザーOFF

      out     PORTB,out_H     ; OUT=H
      nop
      nop
      out     PORTB,out_L     ; OUT=L
      rjmp    DOTDASH        ; DOTDASHへ戻る

```

; * トット、タッチ後の1スペース処理

```

SP1:  cbi     PORTD,BZ          ; 充電開始
      sbi     PORTD,CR        ; BZ OFF
      sbi     DDRD,CR         ; 充電開始
      ldi     cnt1,K1         ; 充電端子OUTに変更
      ldi     cnt2,K2         ; CNT1=K1 充電時間設定L
      ; CNT2=K2 充電時間設定H

SP2:  dec     cnt1              ; 充電完了待ち
      brne   SP2              ; CNT1 マイナス1
      ldi     cnt1,K1         ; CNT1=0まで繰り返す
      dec    cnt2              ; CNT1=K1 充電時間設定L
      brne   SP2              ; CNT2 マイナス1
      ; CNT2=0まで繰り返す

      cbi     DDRD,CR         ; 充電端子INIに戻す
      cbi     PORTD,CR        ; 充電終了

      rcall   FDISP          ; 周波数表示

SP3:  sbis    PINB,SWDOT        ; 放電完了待ち
      ori    memflag,0x04     ; トット ON?
      sbis   PINB,SWDAS       ; Yes! フラグ へ4をORする
      ori    memflag,0x08     ; タッチ ON?
      sbic   ACSR,ACO         ; Yes! フラグ へ8をORする
      ; ACO=L?

```

```

rjmp SP3 ;No!
clr temp ;
out TCCR0, temp ;TMROストップ

```

;次の処理をドット/ダッシュメモリから決める-----

```

cpi dotflag, 1 ;現在ドット?
breq SPDOT ;Yes! SPDOTへ
;No! SPDASHへ

```

SPDASH:

```

sbrc memflag, 0 ;ダッシュ中にドットON? 0x01
rjmp DOT ;Yes! DOTへ

sbrc memflag, 2 ;スペース中にドットON? 0x04
rjmp DOT ;Yes! DOTへ

sbrc memflag, 3 ;スペース中にダッシュON? 0x08
rjmp DASH ;Yes! DASHへ

rjmp MAIN ;メインへ

```

SPDOT:

```

sbrc memflag, 3 ;スペース中にダッシュON? 0x08
rjmp DASH ;Yes! DASHへ

sbrc memflag, 1 ;ドット中にダッシュON? 0x02
rjmp DASH ;Yes! DASHへ

rjmp MAIN ;メインへ

```

```

;*****
;* 電鍵モード
;* キーダウン時キャラを出力する
;*****

```

DENSTART:

```

sbis PINB, SWDOT ;ドット(電鍵) ON?
rjmp DENSTART ;Yes! OFFまで待つ

```

DENKEN:

```

rcall FDISP ;周波数表示

andi out_H, 0xDF ;キャラH出力定数 RY OFF
andi out_L, 0xDF ;キャラL出力定数 RY OFF
out PORTB, out_L ;出力

sbic PINB, SWDOT ;電鍵=L?
rjmp DENKEN ;No! 繰り返す

```

DE1:

```

ldi tmr_K, K3 ;TMRO定数をRdIに設定
out TCNT0, tmr_K ;TMRO定数設定
ldi temp, 3 ;
out TCCR0, temp ;TMROスタート Clock/64
ldi temp, 2 ;
out TIFR, temp ;TMRO TOV0クリア

sbi PORTD, BZ ;プザ-ON
sbr bzf1ug, 1 ;プザ-フラグ 1に設定

ori out_H, 0x20 ;キャラH出力定数 RY ON
ori out_L, 0x20 ;キャラL出力定数 RY ON

```

DE2:

```

out PORTB, out_H ; OUT=H ;プザ-用TMROチェック
in temp, TIFR ;TIFR読み込み
andi temp, 0x02 ;TOV0=H?(オーバー-フロー?)
out PORTB, out_L ; OUT=L
brne DE10 ;Yes! DE10へ ;オーバー-フローした
nop ;No!

```

DE3:

```

out PORTB, out_H ; OUT=H ;KEY DOWN チェック
in temp, PINB ;PORT読み込み
andi temp, 0x40 ;KEY(DOT)=L?
out PORTB, out_L ; OUT=L
breq DE2 ;Yes! DE2へ
nop ;No!

```

```

    rjmp    DE20                ;
;ブザー-ON/OFF処理-----
DE10:
    out     PORTB,out_H        ;      OUT=H                タイマ-再起動
    nop
    out     TCNT0,tmr_K        ;      TMRO定数再設定
    out     PORTB,out_L        ;      OUT=L
    ldi     temp,2
    out     TIFR,temp          ;      TMRO TOVOクリア

    out     PORTB,out_H        ;      OUT=H
    nop
    andi    bzf1ug,1          ;      ブザー-フラグ=1?
    out     PORTB,out_L        ;      OUT=L
    brne    DE12               ;      Yes! DE12へ
    nop

DE11:
    out     PORTB,out_H        ;      OUT=H                前回BZはOFFなのでONにする
    sbr     bzf1ug,1          ;      ブザー-フラグ 1に設定
    nop
    out     PORTB,out_L        ;      OUT=L
    sbi     PORTD,BZ          ;      ブザー-ON

    out     PORTB,out_H        ;      OUT=H
    nop
    nop
    out     PORTB,out_L        ;      OUT=L
    rjmp    DE3               ;      DE3へ戻る

DE12:
    out     PORTB,out_H        ;      OUT=H                前回BZはONなのでOFFにする
    cbr     bzf1ug,1          ;      ブザー-フラグ 0に設定
    nop
    out     PORTB,out_L        ;      OUT=L
    cbi     PORTD,BZ          ;      ブザー-OFF

    out     PORTB,out_H        ;      OUT=H
    nop
    nop
    out     PORTB,out_L        ;      OUT=L
    rjmp    DE3               ;      DE3へ戻る

```

```

;*****
;* 電鍵モードでのセミブレイクイン処理 1ドットぶんのウェイト
;*****

```

```

DE20:
    cbi     PORTD,BZ          ;      充電開始
    sbi     DDRD,CR           ;      BZ OFF
    sbi     PORTD,CR          ;      充電端子OUTに変更
    ldi     cnt1,K1           ;      充電開始
    ldi     cnt2,K2           ;      CNT1=K1 充電時間設定L
                                ;      CNT2=K2 充電時間設定H

DE21:
    dec     cnt1              ;      充電完了待ち
    brne    DE21             ;      CNT1 戻り
    ldi     cnt1,K1           ;      CNT1=0まで繰り返す
    dec     cnt2              ;      CNT1=K1 充電時間設定L
    brne    DE21             ;      CNT2 戻り
                                ;      CNT2=0まで繰り返す

    cbi     DDRD,CR           ;      充電端子INに戻す
    cbi     PORTD,CR          ;      充電終了

DE22:
    sbis    PINB,SWDOT        ;      放電完了待ち
    rjmp    DE1               ;      電鍵=L?
                                ;      Yes! 戻る

    sbic    ACSR,ACO          ;      ACO=L?放電完了?
    rjmp    DE22             ;      No!

    clr     temp              ;
    out     TCCR0,temp        ;      TMROストップ

    rjmp    DENKEN           ;      電鍵モードへ戻る

```

```

;*****
;* 周波数表示サブルーチン
;*****

```

```

FDISP:
    ldi     temp,0x00        ;

```



```

LE4:                                     ;LED4 ON LD543210=>110111
      cbi   PORTD,LD3                   ;LED4,9 ON
      out   PORTB,out_L                 ;出力
      ret

LE3:                                     ;LED3 ON LD543210=>111011
      cbi   PORTD,LD2                   ;LED3,8 ON
      out   PORTB,out_L                 ;出力
      ret

LE2:                                     ;LED2 ON LD543210=>111101
      cbi   PORTD,LD1                   ;LED2,7 ON
      out   PORTB,out_L                 ;出力
      ret

LE1:                                     ;LED1 ON LD543210=>111110
      cbi   PORTD,LDO                    ;LED1,6 ON
      out   PORTB,out_L                 ;出力
      ret

```

```

;*****
;* LEDを消すサブルーチン
;*****

```

```

LEDCLEAR:
      sbi   PORTD,LDO                    ;LED1,6 OFF
      sbi   PORTD,LD1                   ;LED2,7 OFF
      sbi   PORTD,LD2                   ;LED3,8 OFF
      sbi   PORTD,LD3                   ;LED4,9 OFF
      ori   out_H,0x0C                  ;キャリアH出力定数にLD4(PB2),LD5(PB3)を1
      ori   out_L,0x0C                  ;キャリアL出力定数にLD4(PB2),LD5(PB3)を1
      out   PORTB,out_L                 ;出力
      ret

```

```

;*****
;* LEDを順次点灯し、ブザーをピピピと鳴らすサブルーチン
;*****

```

```

LEDCHK:
      rcall LEDCLEAR                    ;LED 消灯
      rcall LE1                         ;LED1 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE2                         ;LED2 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE3                         ;LED3 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE4                         ;LED4 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE5                         ;LED5 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE6                         ;LED6 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE7                         ;LED7 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE8                         ;LED8 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE9                         ;LED9 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE10                        ;LED10 ON
      rcall WAIT1                       ;50ms Wait

      rcall LEDCLEAR                    ;LED 消灯
      rcall LE11                        ;LED11 ON
      rcall WAIT1                       ;50ms Wait

```

```

;*****
;* ピピピ
;*****

```

```

;*****
;PIPIPI:
clr cnt3 ;ピを3回数えるかな
PI1:
clr cnt1 ;プザ -ON/OFFトグルをつくるかな
clr cnt2 ;TMROオハ -70-を繰り返すかな
PI2:
ldi temp,230 ;TMRO定数をRdに設定
out TCNT0,temp ;TMRO定数設定
ldi temp,3
:
:
out TCCR0,temp ;TMROスタート Clock/64
ldi temp,2
:
:
out TIFR,temp ;TMRO TOV0クリア
PI3:
in temp,TIFR ;TIFR読み込み
andi temp,0x02 ;TOV0=H?(オハ -70-?)
breq P13 ;No!

inc cnt1 ;プザ -ONならOFF、OFFならON
sbrc cnt1,0 ;bit0が0ならスキップ
sbi PORTD,0
sbrs cnt1,0
cbi PORTD,0 ;bit0が1ならスキップ

inc cnt2
cpi cnt2,255 ;TMROオハ -70-を255回繰り返す
brlo P12

clr cnt2 ;

PI4:
ldi temp,230 ;TMRO定数をRdに設定
out TCNT0,temp ;TMRO定数設定
ldi temp,3
:
:
out TCCR0,temp ;TMROスタート Clock/64
ldi temp,2
:
:
out TIFR,temp ;TMRO TOV0クリア
PI5:
in temp,TIFR ;TIFR読み込み
andi temp,0x02 ;TOV0=H?(オハ -70-?)
breq P15 ;No!

inc cnt2
cpi cnt2,255 ;TMROオハ -70-を255回繰り返す
brlo P14

inc cnt3
cpi cnt3,3 ;ピを3回繰り返す
brlo P11

ret

;*****
;* ウェイトサブルーチン
;*****
WAIT1:
ldi cnt1,255
ldi cnt2,255
ldi cnt3,2
WA1:
dec cnt1
brne WA1
ldi cnt1,255
dec cnt2
brne WA1
ldi cnt1,255
ldi cnt2,255
dec cnt3
brne WA1
ret

;**** End of File *****

```