

# ICチェッカーの製作

## — 無接点シーケンス制御と電子回路で使える小品 —

秋田センター（秋田職業能力開発促進センター） 千葉 富雄

### 1. はじめに

本作品はデジタルICが良品か不良品かの判定を、だれでも簡単に行える試験器です。

当センターの電気設備科では「無接点シーケンス制御」を2ユニット勉強します。実技課題はブレッドボードを使用して実習回路の組み立てを行っています。受講生が回路を組み立てても作動しないとき、配線に原因があるのか、それともICが壊れたのかわからないことがあります。電子回路に初めて触る方のために部品の取り扱いには注意していますが、過去にはICそのものが壊れていたこともあります。ICが良品かどうかわかっているならば、配線のチェックに集中できることになります。

ICの良品、不良品を簡単に判別できれば有用であると思ったのが今回のICチェッカーを作った動機です。

### 2. 動作原理

デジタルICは与えられた入力信号に応じて、結果を出力します。例えばTTLのAND素子は2本の入力ピンに5Vを与えれば、出力ピンに5Vが出ます。電源とテストがあれば、この試験をブロック（IC内の素子1個）ごとに何回か繰り返すことで、1個のICの良否判定ができます。

本作品は「人が手作業で行う試験」を、マイコン



写真1 作品の外観

が決められた順番に、かつ瞬時に行うものです。不良ブロックの判定もできるようにしました。被測定ICの入力信号と出力信号、そしてLEDの表示はすべてマイコンが管理しています。

ICをチェックする部分の動作例を、図2のフローチャートに示します。

### 3. 対応するICについて

チェックできるICは次の4種類です。

1. SN7400 (NAND)
2. SN7404 (NOT)
3. SN7408 (AND)
4. SN7432 (OR)

この1～4の数字を便宜上IC番号とします。

また、IC内のブロック番号はピン番号の若い順

から付与します。

さらに、ゼロプレッシャー ICソケットの空き部分を使って、タイマー ICである555の動作チェックができます。ユニバーサル基板の一部に555用の発振回路を構成しています。555をソケットに差し込んで電源スイッチを入れると、ICが良品の場合はLEDが点滅します。LEDが点滅しない場合は不良品と判断します。こちらは単独で動作します。

#### 4. 使用部品について

- マイコン (PIC16F628A)
- ICソケット (18ピンDIP)
- ゼロプレッシャー ICソケット (28ピン)
- タクトスイッチ (5個)
- トグルスイッチ (電源スイッチ)
- LED (4個)
- ユニバーサル基板

マイコンとしてPIC16F628Aを選定した理由は次のとおりです。

- ・ CPUクロック発振回路を内蔵している
- ・ 電源ピンを除く全ピンを入出力に使える

#### 5. 回路図

図1の回路図のように、被測定ICの電源ピンを除く全ピンをPICと直結しています。

PICの入出力ピンはプルダウンまたはプルアップをして、被測定ICの入力ピンが開放状態とならないようにしました。

ICへの供給電圧は5Vです。本作品はACアダプタを使用していますが、可搬性を考えて乾電池を電源としてもよいと思います。

#### 6. 制作

ユニバーサル基板に組立て、ケースに収めました。部品の配置はICの抜き差しがやりやすいことと、スイッチを操作しやすいことに考慮しました。(写真1参照)

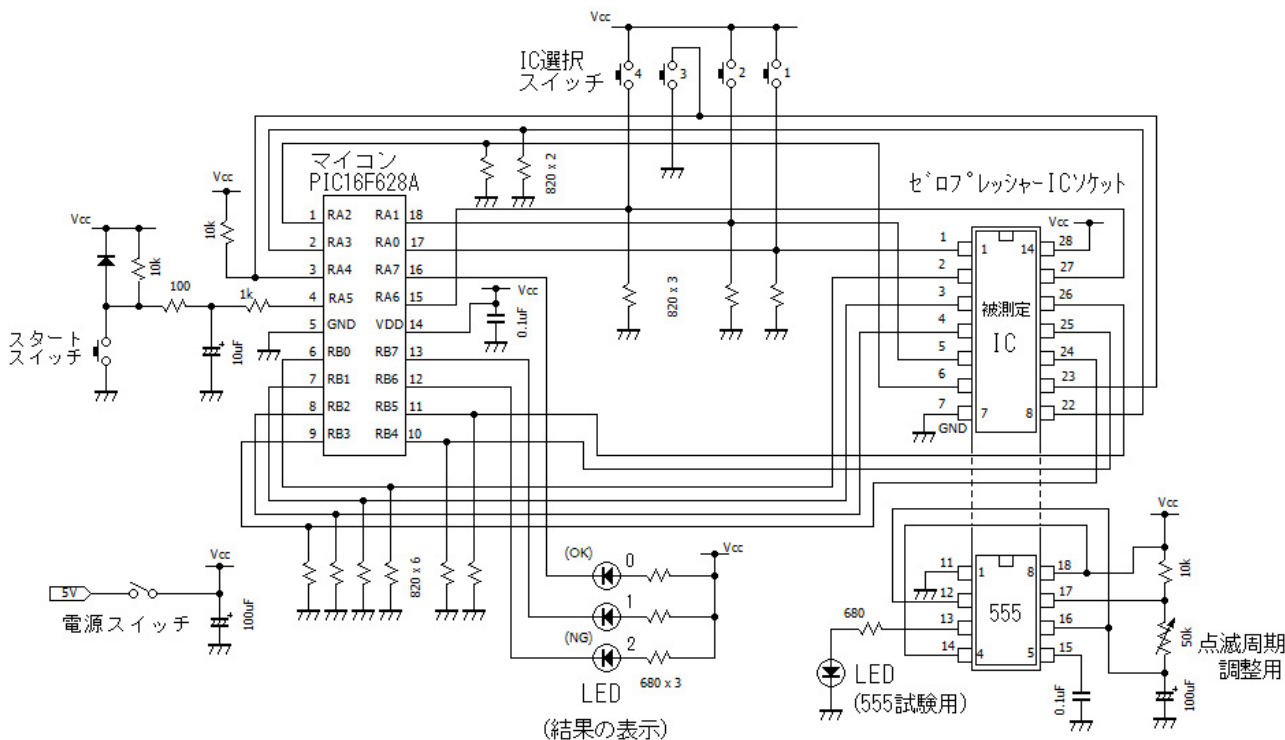


図1 回路図

## 7. 使用方法

ICの抜き差しは電源を切った状態で行います。

### 7.1 7400 / 7404 / 7408 / 7432 のテスト

- (1) テストするICの1番ピンを上側にして、ゼロプレッシャICソケットの「上端」に合わせて挿入ソケットのレバーをロック側に倒す。
- (2) 電源スイッチを入れる。  
(LEDが3個点滅する)
- (3) IC選択スイッチを押す。  
(IC番号が2進数でLEDに表示される)
- (4) スタートスイッチを押す。
- (5) 結果がLEDに表示される。

- ① OKが点灯したら「良品」

	2	1	0
(LED)	○	○	●
	NG		OK
	(点灯)		

- ② NGが点灯したら「不良品」

	2	1	0
(LED)	●	○	○
	NG		OK
	(点灯)		

NGが点灯し、不良ブロックを知りたいときは続けて次の操作をする。

- ③ スタートスイッチを押す。
- ④ 最初の不良ブロック番号がLEDに2進数で表示される。

例えば2つ目のブロックが不良のときは

	2	1	0
(LED)	○	●	○
	(点灯)		

このように2進数で2を表示する。

- ⑤ スタートスイッチを押す。
- ⑥ 次の不良ブロック番号がLEDに2進数で表示される。

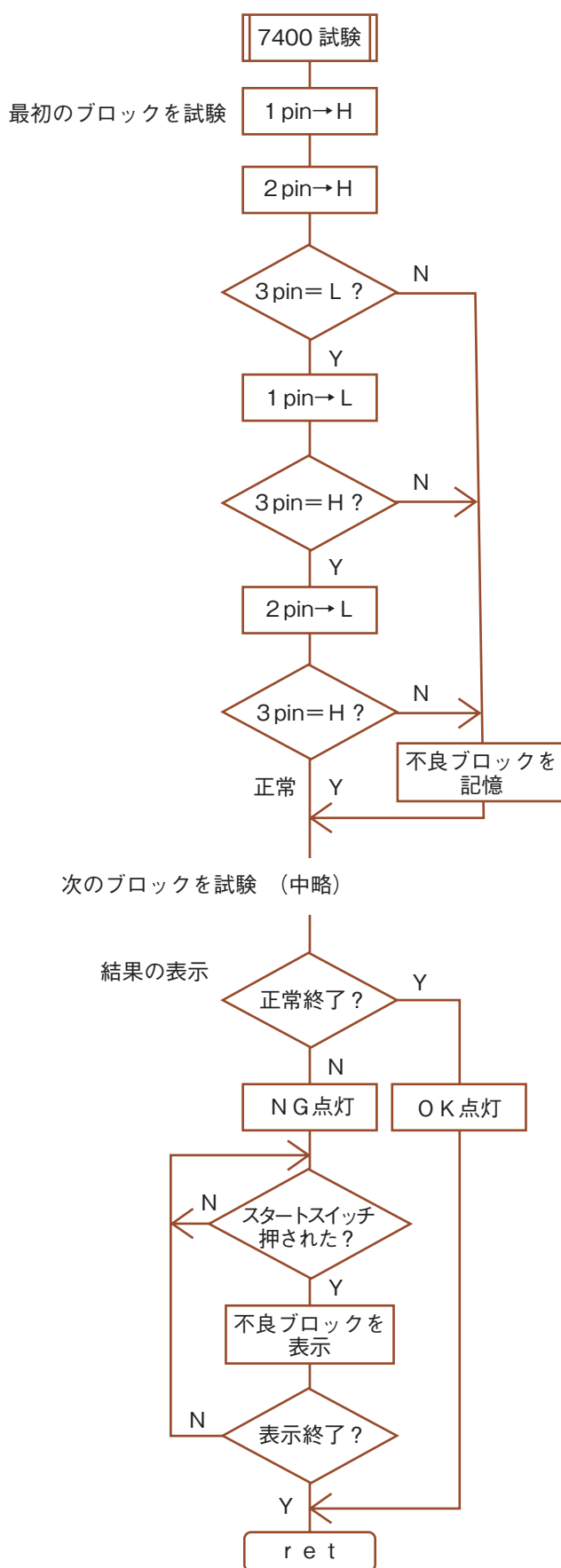


図2 フローチャート (個別処理7400の例)

(⑤と⑥をLEDが消灯するまで繰り返す)

⑦ LEDが消灯したら不良ブロック表示の終了となる。

⑧ ③に戻る (繰り返し)

(6) 終了

電源スイッチを切ってICを抜く。

## 7.2 タイマIC555のテスト

(1) テストする555の1番ピンを上側にして、ゼロプレッシャ ICソケットの「下端」に合わせて挿入する。

(2) 電源スイッチを入れる。

(3) 結果が出る。

① LEDが点滅すると「良品」  
(半固定VRで点滅周期が変わる)

② LEDが点滅しないと「不良品」

(4) 終了

電源スイッチを切ってICを抜く。

## 8. プログラムリスト (抜粋)

```

; -----ICchecker. src-----
include '16f628a.inc'
FUSES _INTOSC_OSC_NOCLKOUT
FUSES _WDT_OFF
FUSES _PWRTE_ON
FUSES _BODEN_ON
FUSES _MCLRE_OFF
FUSES _CP_OFF
FUSES _LVP_OFF
; -----各種定義-----
sw1      equ    ra.0    ;IC選択スイッチ 1
sw2      equ    ra.1    ;IC選択スイッチ 2
sw3      equ    ra.4    ;IC選択スイッチ 3
sw4      equ    ra.6    ;IC選択スイッチ 4
start_sw equ    ra.5    ;スタートスイッチ
led1     equ    ra.7    ;LED1
led2     equ    rb.7    ;LED2
led3     equ    rb.6    ;LED3
OK       equ    ra.7    ;OK表示LED
NG       equ    rb.6    ;NG表示LED
flashtm  equ    0a0h    ;LED点滅初期値

```

```

pin1     equ    ra.0    ;7400,7408,7432
pin2     equ    rb.0    ;で使用
pin3     equ    rb.1
pin4     equ    rb.2
pin5     equ    ra.1
pin6     equ    ra.2
pin8     equ    ra.3
pin9     equ    ra.4
pin10    equ    rb.3
pin11    equ    rb.4
pin12    equ    rb.5
pin13    equ    ra.6
pin1_04  equ    ra.0    ;7404で使用
pin2_04  equ    rb.0
pin3_04  equ    rb.1
pin4_04  equ    rb.2
pin5_04  equ    ra.1
pin6_04  equ    ra.2
pin8_04  equ    ra.3
pin9_04  equ    ra.4
pin10_04 equ    rb.3
pin11_04 equ    rb.4
pin12_04 equ    rb.5
pin13_04 equ    ra.6
org      20h
res7400  ds     1      ;不良ブロック記憶用
res7404  ds     1
res7408  ds     1
res7432  ds     1
tm1      ds     1      ;タイマ用カウンタ
tm2      ds     1
tm3      ds     1
org      0
goto    start
;-----プログラム本体-----
org     4
start   setb   CM0      ;628特有の設定
        setb   CM1      ;
        setb   CM2      ;
        setb   TOCS     ;ここまで
        mov    !RA, #01111111b
        mov    !RB, #00001111b

```

```

:loop    call    led_on      ;LED3個点滅
        call    wait0
        call    led_off
        call    wait1
        jmp     :loop

wait0    mov     tm1, #0a0h
:wait01  mov     tm2, #0
:wait02  jb     sw1, i7400    ;個別処理へ
        jb     sw2, i7404
        jnb    sw3, i7408
        jb     sw4, i7432
        djnz   tm2, :wait02
        djnz   tm1, :wait01
        ret

wait1    mov     tm1, #0a0h
:wait01  mov     tm2, #0
:wait02  jb     sw1, i7400    ;個別処理へ
        jb     sw2, i7404
        jnb    sw3, i7408
        jb     sw4, i7432
        jnb    start_sw, :fin
        djnz   tm2, :wait02
        djnz   tm1, :wait01
        ret

:fin     call    led_off      終了処理
:halt    jmp     :halt
;-----7400の処理-----
i7400    ;スタート待ち
        call    led_off
:lp      clrb    led1
        mov     tm1, #flashtm
:wt1     mov     tm2, #0
:wt2     jnb    start_sw, :test
        djnz   tm2, :wt2
        djnz   tm1, :wt1
        setb   led1
        mov     tm1, #flashtm
:wt3     mov     tm2, #0
:wt4     jnb    start_sw, :test
        djnz   tm2, :wt4
        djnz   tm1, :wt3
        jmp    :lp

:test    jnb    start_sw, :test
        call    wait0
        clr    res7400
        mov    !RA, #00101100b
        mov    !RB, #00010010b
        call    ledoff

:blk1    setb   pin1        ;1pin→0
        setb   pin2        ;2pin→0
        jb     pin3, :b1    ;3pin= 1 か?
        clrb   pin1        ;1pin→0
        jb     pin3, :b1    ;3pin= 1 か?
        clrb   pin2        ;2pin→0
        jb     pin3, :blk2 ;3pin= 1 か?
:blk2    setb   res7400.1   ;不良ブロック記憶

:blk3    setb   pin4        ;4pin→0
        setb   pin5        ;5pin→0
        jb     pin6, :b2    ;6pin= 1 か?
        clrb   pin4        ;4pin→1
        jb     pin6, :blk3  ;6pin= 0 か?
:blk4    setb   res7400.2   ;不良ブロック記憶

:blk5    setb   pin9        ;9pin→0
        setb   pin10       ;10pin→0
        jb     pin8, :b3    ;8pin= 1 か?
        clrb   pin9        ;9pin→1
        jb     pin8, :blk4 ;8pin= 0 か?
:blk6    setb   res7400.3   ;不良ブロック記憶

:blk7    setb   pin12       ;12pin→0
        setb   pin13       ;13pin→0
        jb     pin11, :b4   ;11pin= 1 か?
        clrb   pin12       ;12pin→1
        jb     pin11, :report ;11pin= 0 か?
:blk8    setb   res7400.4   ;不良ブロック記憶

:report  mov     RA, #10000000b
        mov     RB, #11000000b
        and    res7400, #00011110b
        jnz    :err_disp
        clrb   OK          ;OK点灯

```

```

:fin      jmp      :fin      ;正常終了

:err_disp clrb   NG          ;NG点灯
          ;不良ブロックの表示
:err1     jb     start_sw,:err1
          jnb    res7400.1,:err2
          call   led_off
          clrb   led1        ;ブロック1表示
:err01    jnb    start_sw,:err01
:err2     jb     start_sw,:err2
          jnb    res7400.2,:err3
          call   led_off
          clrb   led2        ;ブロック2表示
:err02    jnb    start_sw,:err02
:err3     jb     start_sw,:err3
          jnb    res7400.3,:err4
          call   led_off
          clrb   led2        ;ブロック3表示
          clrb   led1
:err03    jnb    start_sw,:err03
:err4     jb     start_sw,:err4
          jnb    res7400.4,:err_end
          call   led_off
          clrb   led3        ;ブロック4表示
:err04    jnb    start_sw, :err04

          ;不良ブロックの表示終了
:err_end  jb     start_sw,:err_end
          call   led_off     ;LED消灯
:err05    jnb    start_sw,:err05
          jmp    :err1       ;繰り返し
(以下省略)
; -----
          ; 表示用LED 3個点灯
led_on    clrb   led1
          clrb   led2
          clrb   led3
          ret
          ; 表示用LED 3個消灯
led_off   setb   led1
          setb   led2
          setb   led3
          ret
; -----

```

アセンブラはTechTools社のCVASM16を使用しました。CVASM16はZ80のニーモニックにPICのビット操作命令を追加したような仕様なので、Z80を知っている方はすぐに使えると思います。

TechTools社のホームページから無料でダウンロードできます。Microchip社のMPASMで作成する場合は、CVASM16のヘルプファイルにMPASMとの対応表が記載されていますので参考にしてください。

## 9. 開発環境

Windows パソコン  
エディタ (K2Editor) フリーソフト  
アセンブラ (TechTools社 CVASM16) フリーソフト  
作図ソフト (水魚堂BSch3V) フリーソフト  
PICライター (秋月電子PICプログラマー ver. 4)

## 10. おわりに

今回は被測定ICとして、授業で使用する種類に限定して制作しましたが、考えさえつかめば他の種類のICにも応用ができます。カウンタICなどのチェックも可能です。同じ原理で、LANケーブルやRS-232Cケーブル等のチェックも可能です。

本作品の欠点は、ICへの入出力をプログラムで1つひとつ記述していることです。そのためチェックしたいICの種類が変わったり、ピン番号が変わった場合、プログラムをそれに合わせて変更する必要があります。この点は授業で使用するICに特化して作る、という考え方でよいと思います。本作品はスイッチを3個、LEDを3個を使いますので、スイッチの押し方を2進数に置き換えると最大8種類のICまで対応できます。

ご不明の点がありましたら秋田センターまでご連絡ください。