Aplicaţii cu PIC – uri

Aplicaţia numărul 3 se va referi la schema din Figura nr. 1.

1. Exemplul 1

În continuare se va da un program ce realizează aprinderea şi stingerea unui LED cu timpi inegali. Acest lucru a fost posibil prin introducerea celei de-a doua subrutine de întârziere. Programul este o dezvoltare a celui prezentat laboratorul trecut.

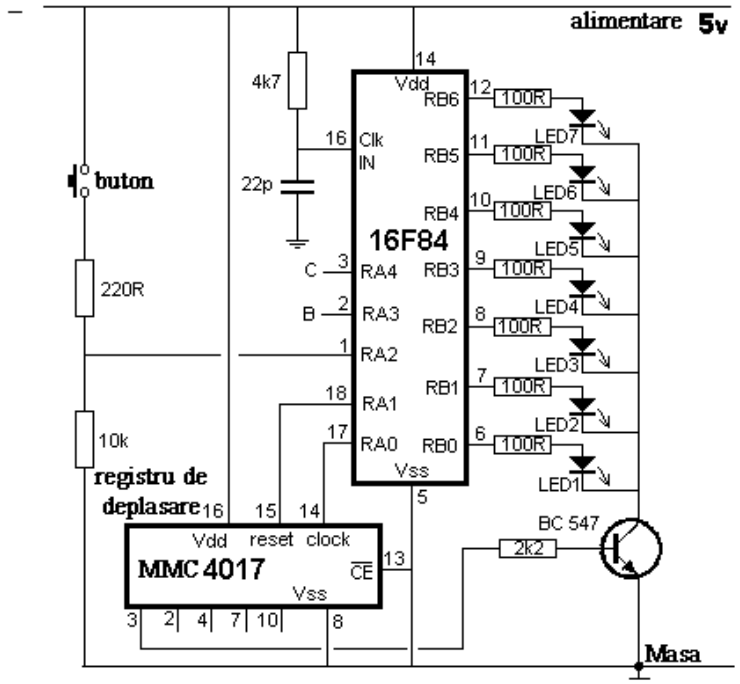


Fig. 1 schema de test

```
include   <p16F84.inc>    ;foloseste
          ;trasaturile microcontrolerului
          ;pic16F84
ORG 0000H
                          ;inscrie
          ;memoria program de ladresa
          0000H
Start
          BSF 03,5        ;seteaza bitul 5
          ;dela     ;adresa 03H , adica pe
          RP0
          MOVLW 00h       ;muta 00 in
                          ;Acumulator
          MOVWF 06h               ;muta din acumulator in registrul TRISB
          BCF 03,5                ;reseteaza bitul 5 dela adresa 03H , adica pe RP0
ionel
          MOVLW 01h       ;muta 01 in Acumulator
          MOVWF 06h       ;muta in PORTB valoarea din Acumulator; aprinde LED-ul conectat la LSB din
          ;PORTB
          CALL Delay1     ;apeleaza o intarziere
          MOVLW 00h       ;muta 00 in Acumulator
          MOVWF 06h       ;stinge LED-ul prin mutarea in PORTB a valorii 00 din Acumulator
          CALL Delay2     ;apeleaza o intarziere
          GOTO ionel              ;du-te la eticheta ionel

Delay1
          MOVLW .13               ;muta constanta ZECIMALA 13, in Acumulator
          MOVWF 1Ah               ;muta din Acumulator la adresa 1A hexa
Delay10
          DECFSZ 1Bh,1
                    ;daca pana acuma nu am lucrat deloc cu locatia 1BH, inseamna ca ea are valoarea FFH, si de aici
                    se decrementeaza ...
          GOTO Delay10
          DECFSZ 1Ch,1
          GOTO Delay10
          DECFSZ 1Ah,1
          GOTO Delay10
          RETURN
Delay2
          MOVLW .20               ;muta constanta ZECIMALA 20, in Acumulator
          MOVWF 1Ah               ;muta din Acumulator la adresa 1A hexa
Delay20
```

```
        DECFSZ 1Bh,1   ;daca pana acuma nu am lucrat deloc cu locatia 1BH, inseamna ca ea are valoarea FFH,
        si de aici se decrementeaza ...
        GOTO Delay20
        DECFSZ 1Ch,1
        GOTO Delay20
        DECFSZ 1Ah,1
        GOTO Delay20
        RETURN
END                                          ;sfarsit de program
```

2. Exemplul 2

Următoare aplicaţie are se referă la o scanare de port. Prin rotiri stânga sau dreapta, un nivel 1 logic (lumina unui LED) va balea portul într-un sens şi în celălalt. Să presupunem că nivelul logic „merge" spre stânga, în momentul când ajunge la ultimul bit, acest lucru este detectat şi se va da comandă inversă pentru ca „lumina" să „meargă" spre dreapta. Acelaşi lucru se va întâmpla şi la cealaltă extremitate a portului; astfel încât „lumina" va rămâne blocată în interiorul portului.

```
;include   <p16F84.inc> ;foloseste trasaturile microcontrolerului  pic16F84
ORG 0000H               ;inscrie memoria program de la adresa 0000H
Start
  BSF 03,5              ;bank 1 de memorie
  MOVLW 00h             ;incarca 00 in acumulator
  MOVWF 05h             ;incarca TRIS A
  MOVWF 06h             ; incarca TRIS B
  BCF 03,5              ;bank 0 de memorie
  BCF 03h,0             ;sterge flagul Carry
  MOVLW 01h             ;incarca 01 in W
  MOVWF 06h             ;aprinde primul LED
  CALL Delay            ;intГrziere
stanga
  RLF 06h,1             ;roteste spre stanga
  CALL Delay            ;intГrziere
  BTFSS 06h,7           ;a ajuns ôluminaö la capat ?
  GOTO stanga           ;NU , atunci repeta
dreapta
  RRF 06h,1         ;DA, a ajuns ôluminaö la ultimul bit. Acum roteste dreapta
  CALL Delay            ;intarziere
  BTFSS 06h,0           ; a ajuns ôluminaö la capat ?
  GOTO dreapta          ;NU
  GOTO stanga           ;DA, repeata intregul ciclu

Delay
  MOVLW 03              ;rutina de intarziere
  MOVWF 1Ah
Delay1
  DECFSZ 1Bh,1
  GOTO Delay1
  DECFSZ 1Ch,1
  GOTO Delay1
  DECFSZ 1Ah,1
  GOTO Delay1
  RETURN

END                     ;sfarsit de program
```

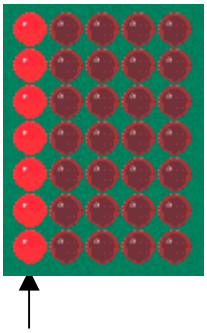
```
$include (reg552.inc)
org 0000H

START:

        MOV P3,#00H
UP:

        MOV P3,#01H
        CALL DELAY
UP1:
        MOV A,P3
        RL A
        MOV P3,A
        CALL DELAY
        JNB P3.7,UP1
UP2:
        MOV A,P3
        RR A
        MOV P3,A
        CALL DELAY
        JNB P3.0,UP2
        JMP UP1
DELAY:
        MOV R7,#1
        MOV R6,#1
        MOV R5,#1
DELAY1:
        DJNZ R5,DELAY1
        DJNZ R6,DELAY1
        DJNZ R7,DELAY1
        RET
END
```

3. Exemplul 3

Următorul exemplu „mută" o coloană „aprinsă" când spre stânga, când spre dreapta. Acest lucru este posibil folosind ambele porturi ale microcontrolerului, Portul A este folosit pentru a da impulsuri Registrului de Deplasare realizat cu MMC4017. Mai întâi RDR (registrul de deplasare) este resetat , cu un impuls format la pinul de port RA1. Impulsurile de deplasare ale RDR – ului sunt aduse de la pinul RA0. (Observaţi ca $\overline{CE}$ este legat la MASA, ca urmare deplasarea bitului din RDR are loc NUMAI într-un singur sens), ca urmare este foarte uşor de a „muta" coloanele într-un singur sens; în celălalt sens este mai greu, pentru că nu mai pot utiliza opţiunile RRF şi RLF, ci rotirile îmi sunt date doar de RDR. Ca sa „mut" coloanele în sens invers, resetez RDR (in acest moment ar trebui să fie luminoasă prima coloană) şi daca in acest moment, aplic 3 impulsuri la pinul de clock al RDR – ului, ma voi afla pe coloana nr. 4. Abia în acest moment voi aprinde coloana – si apoi o voi stinge – cu timp de aprindere si de stingere egal. Ca să aprind coloana nr. 3, resetez iar RDR – ul, dar de această data, am să-i dau doar 2 impulsuri de clock, şi apoi am să aprind ce-e de-a 3 coloană. ş.a.m.d.

```
        ;include   <p16F84.inc>   ;foloseste trasaturile microcontrolerului  pic16F84
        ORG 0000H                 ;inscrie memoria program de la adresa 0000H
Start
        BSF 03,5          ;selectez pagina 1 din SFR
        MOVLW 00h         ;muta 00 în acumulator
        MOVWF 05h         ;setaza portulA sa fie de iesire
        MOVWF 06h         ; setaza portulB sa fie de iesire
        BCF 03,5          ; selectez pagina 0 din SFR

Shift   MOVLW 05
        MOVWF 19h         ;încarca 5 la adresa 19H , (pentru ca am 5 coloane)
        BSF 05h,1         ; formeaza impuls de  Reset 4017
        BCF 05h,1         ;

Shift1  MOVLW 0FFh
        MOVWF 06h         ;aprinde toate LED - urile
        CALL Delay        ; apeleaza întârzierea
        MOVLW 00          ;
        MOVWF 06h                 ; stinge LED – urile
        CALL Clock        ;apeleaza rutina de clock pentru MMC4017 (RDR)
        DECFSZ 19h,1      ;decrementeaza registrul 19H (unde am încarcat nr. de coloane (5)).
        GOTO Shift1               ;daca nu ai ajuns la ultima coloana, repeta!

        MOVLW 03h         ;daca am ajuns aici înseamna ca suntem pe calea de întoarcere si trebuie sa aprindem
                          ;coloana nr. 4
        MOVWF 19h         ;încarc 3 în regitrul 19H (am nevoie doar de 3 impulsuri ca sa ajung la coloana a 4-a)
AAA     CALL Back
        DECFSZ 19h,1      ;prin decrementare, ajung la coloanele 4, 3, 2. Coloana 1 este la RESET.
        GOTO AAA
        GOTO Shift

Back    BSF 05h,1
        BCF 05h,1         ;impuls de RESET pentru MMC4017
        MOVLW 00h
        MOVWF 06h         ;stinge toate LED – urile
        MOVF 19h,0        ;muta din regitrul 19H în Acumulator  (atentie la acel 0)
        MOVWF 18h         ;copie din Acumulator în registrul intermediar 18H
```

```
BBB     CALL Clock      ;cu ajutorul registrului 18H, se formeaza impulsurile de clock pentru
                        ;MMC4017, se dau
        DECFSZ 18h,1    ; atâtea impulsuri cât este necesar pentru a ajunge pe (coloana +1) dorita
        GOTO BBB
        MOVLW 0FFh
        MOVWF 06h       ;aprinde toate LED – urile
        CALL Delay      ;apeleaza o rutina de întârziere
        RETURN          ;întoarce-te la eticheta AAA

Clock   BCF 05,0        ;Clock pentru MMC4017
        NOP             ;MMC4017 este un pic mai lent, si de aceea se lasa un pic de
                        ;întârziere
        BSF 05,0
        RETURN

Delay   MOVLW 03        ;rutina de întârziere
        MOVWF 1Ah
Delay1  DECFSZ 1Bh,1
        GOTO Delay1
        DECFSZ 1Ch,1
        GOTO Delay1
        DECFSZ 1Ah,1
        GOTO Delay1
        RETURN

        END             ;sfârsit de program
```



Coloana 1