

Aplicație

În cele ce urmează este prezentat un program de comandă pentru Invertorul monofazat complet comandat. (softul este același și pentru Invertorul monofazat semicomandat). La ieșirile P3.0 și P3.1 se obțin semnale de comandă PWM (MID) natural pentru cele două diagonale ale punții monofazate.

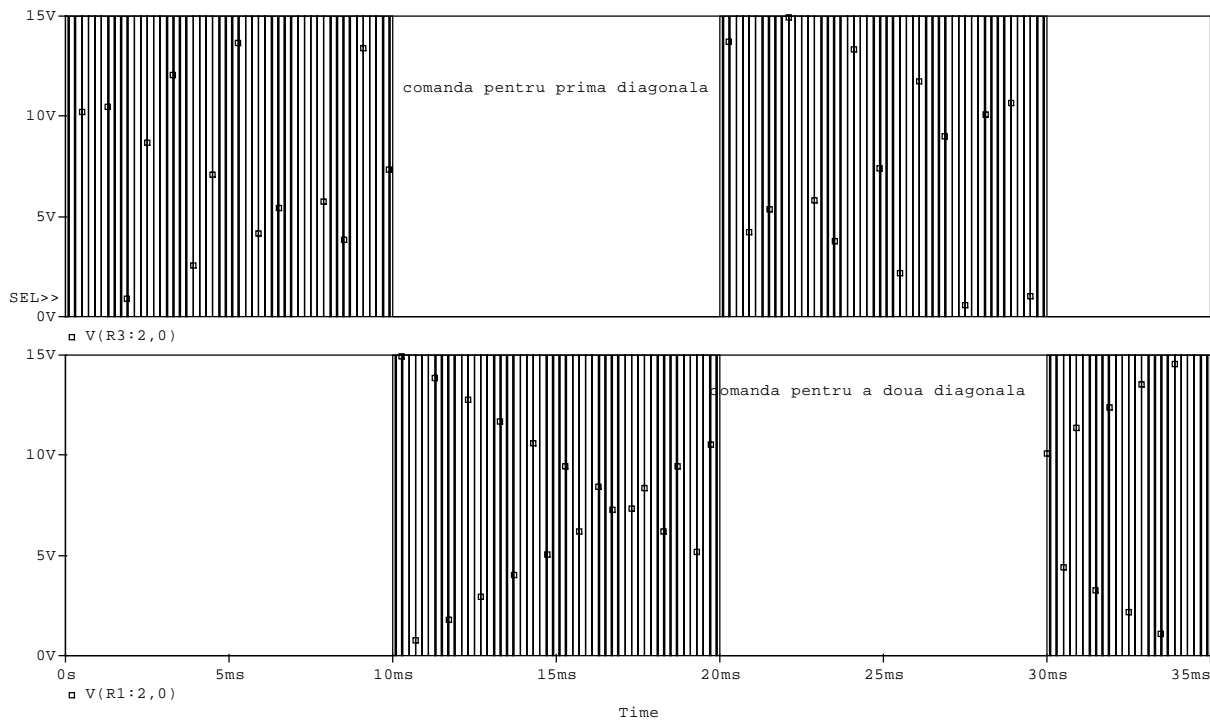


Fig. 1 Se dau formele de undă de la ieșirea pinilor P3.0 și P3.1

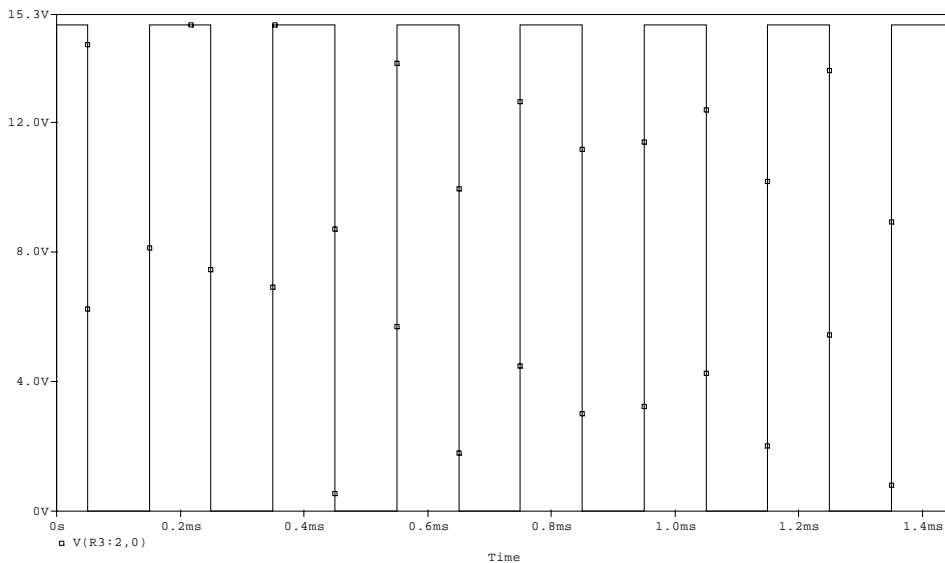


Fig. 2 Alăturat, este prezentat mai în detaliu, impulsurile de comandă pentru una din diagonale.

Funcționare d.p.d.v. electronic: într-o semiperioadă (10 ms, la fel ca o semiperioadă a tensiunii de la rețea!!!) comand o singură diagonală de tranzistoare din puntea monofazată (cealaltă diagonală rămâne necomandată) cu o frecvență de 10KHz (după cum se obs. din Fig. 2). Deci pinul de port P3.0, la momentul inițial se va seta (1-logic), apoi se merge într-o întârziere $t_1 = 50$ microsecunde ($f = 10\text{KHz}$, rezultă, $t = 1 / f$, adică $t = 100$ microsecunde), când se revine, se „șterge” (0-logic), pinul de port P3.0, și iarăși se trimite într-o întârziere de 50 microsecunde. ($t = t_1 + t_1 = 100$ microsecunde). Acest lucru se repetă pe tot decursul semiperioadei de 10ms. După ce au trecut 10 ms, nu se mai comandă acest pin de port (P3.0) și se va comanda după aceleași principii pinul de port P3.1, astfel încât să se obțină formele de undă din Fig. 1. Temporizarea semiperioadei de 10ms, este realizată cu ajutorul Timerului T0.

Funcționare soft: inițial este „pregătit” Timerul T0 să lucreze în modul 1 de funcționare (Timer/numărător pe 2 octeți) , aceștia fiind încărcăți cu niște valori astfel încât, acesta, după 10milisecunde să semnalizeze „depășire” intrând într-o rutină de întrerupere.

Calculul valorilor de preîncărcare a Timerului T0:

Timerul T0 se „umple” atunci când ajunge în starea FFFFH, adică 65535 în zecimal. Din această valoare scad cele 10.000 de microsecunde. (65535 sunt tot microsecunde deoarece T0 se incrementează la fiecare microsecundă!). $65535 - 10000 = 55535$, adică D8EFH scris în format hexa. Ca urmare, TH0 = D8H și TL0 = EFH.

Cu ajutorul Timerului T0 formez momentele de schimbare de semialternanță. (din 10 în 10 ms.) În program, mai este folosit flagul Carry, astfel încât , dacă $C = 0$, sunt pe „semialternanța” 1, realizată cu ajutorul etichetei „salt0”, iar dacă $C = 1$, sunt pe „semialternanța” 2, realizată cu ajutorul etichetei „salt1”. Bascularea flagului Carry are loc la fiecare întrerupere (prin complementare), întrerupere ce are loc la fiecare 10 milisecunde.

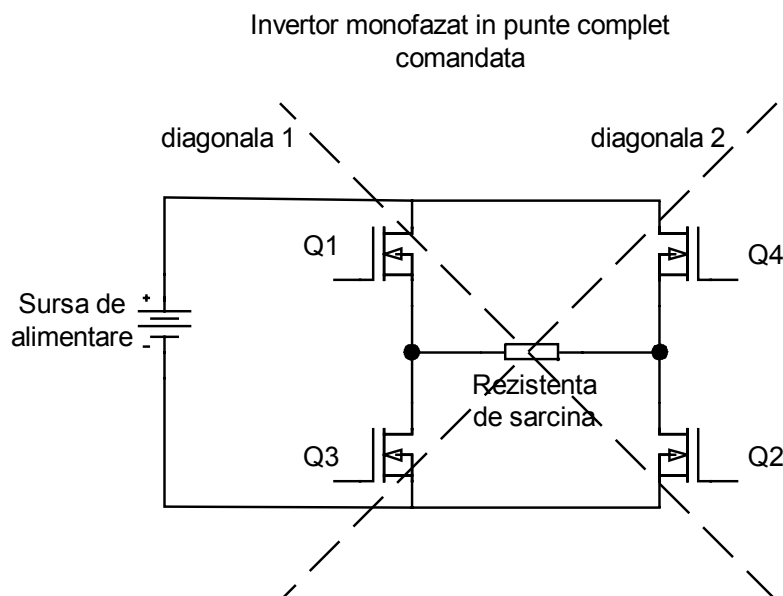


Fig. 3

name comanda_PWM_natural_invertor_monofazat

```
ORG 0000H    ; ADRESA MEMORIE
JMP START    ;
```

```
ORG 000BH    ; ADRESA INTRERUPERE T0
JMP INIT     ; SALT LA INTRERUPERE
```

```
START: MOV P3, #00H        ;STERG PORTUL P3
      MOV TL0, #0EFH       ;INCARC OCTETUL INFERIOR T0
      MOV TH0, #0D8H       ;INCARC OCTETUL SUPERIOR T0
      MOV IE, #02H         ;ACTIVEZ INTRERUPERE T0
      SETB EA               ;ACTIVEZ GLOBAL INTRERUPERILE
      MOV TMOD, #01H       ;SETEZ T0 PE 16 BITI CA TIMER
      CLR C                 ;STERG CARRY
      SETB TR0              ;PORNESC T0
```

```
SALT0: SETB P3.0           ;ACTIVEZ 1 LOGIC LA PINUL P3.0
      CALL DELAY           ;ASTEPT
      CLR P3.0             ;STERG BITUL P3.0
      CALL DELAY           ;ASTEPT
      JC SALT1             ;VERIFIC DACA A AVUT LOC O INTRERUPERE
      JMP SALT0
```

```
SALT1: SETB P3.1           ;ACTIVEZ 1 LOGIC LA PINUL P3.1
      CALL DELAY           ;ASTEPT
      CLR P3.1             ;STERG BITUL P3.1
      CALL DELAY           ;ASTEPT
      JNC SALT0            ;VERIFIC DACA A AVUT LOC O INTRERUPERE
      JMP SALT1
```

```
INIT:                ;IN ACEST MOMENT, AU TRECUT 10MS
      CLR TF0              ;STERG FLAGUL INTRERUPERE T0
      CPL C                ;COMPLEMENTEZ CARRY
      MOV TL0, #0EFH       ;INCARC OCTETUL INFERIOR T0
      MOV TH0, #0D8H       ;INCARC OCTETUL SUPERIOR T0
      RETI
```

```
DELAY:      MOV R0, #22    ;REALIZEZ INTERZIEREA DE 50uS
DELAY1:     DJNZ R0, DELAY1
           RET
```

END

2. Convertorul Analog Digital (de la microcontrolerul 80C552- Philips)

Circuitul analogic de intrare constă dintr-un multiplexor analogic de 8 biți și un convertor analog digital cu aproximații succesive de 10 biți, direct binar. Tensiunea analogică de referință și tensiunea analogică de alimentare sunt conectate prin intermediul unor pini separați. Conversia necesită 50 cicluri mașină, respectiv 37.5 μs la o frecvență de 16 MHz a oscilatorului, 25 μs la 24 MHz. Tensiunea de intrare variază de la 0 la + 5V.

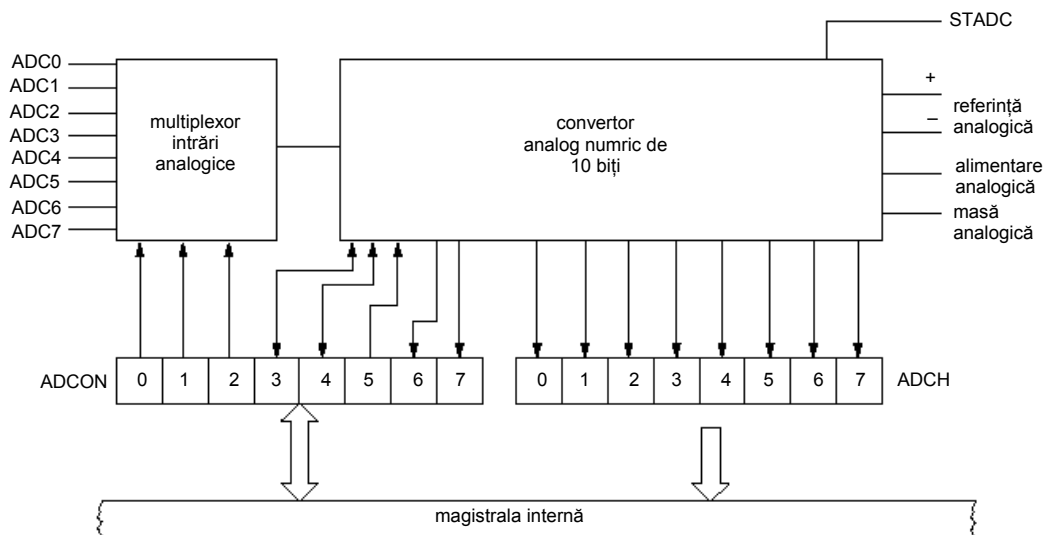


Fig. 4 Prezintă o schema funcțională a circuitului analogic de intrare.

Conversia analog-digitală. Figura 5 prezintă elementele unei conversii analog digitale cu aproximații succesive. Convertorul A/D conține un convertor numeric-analogic care convertește conținutul unui registru cu aproximații succesive într-o tensiune V_{DAC} ce este comparată cu tensiunea analogică de intrare (V_{in}).

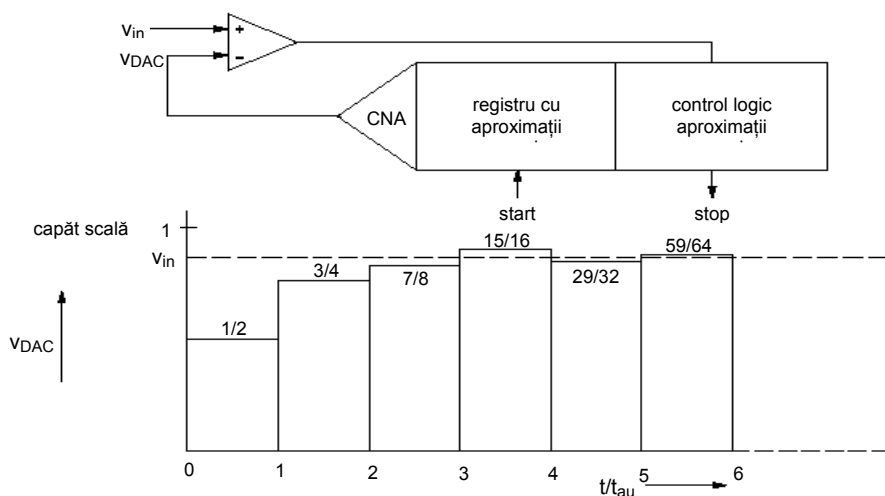


Fig. 5

Ieșirea comparatorului este livrată logicii de control care controlează registrul de aproximații succesive. O conversie este inițiată prin setarea **ADCS** în registrul **ADCON**. **ADCS** poate fi setat numai software; sau fie hardware fie software.

Modul de start numai software este selectat când bitul de control **ADCON.5 (ADEX)=0**. O conversie este declanșată atunci prin setarea bitului de control **ADCON3 (ADCS)**. Modul de declanșare hardware sau software este selectat când **ADCO.5 = 1**, iar o conversie poate fi declanșată prin setarea **ADCON.3** ca mai sus sau prin aplicarea unui front crescător la pinul **STADC**. Când o conversie este declanșată prin aplicarea unui front crescător la **STADC** trebuie aplicat un nivel **LOW** pentru cel puțin un ciclu mașină, urmat de un nivel **HIGH** pentru cel puțin un ciclu mașină.

Controlul logic al aproximării succesive setează mai întâi cel mai semnificativ bit și șterge toți ceilalți biți din registrul cu aproximații succesive (10 0000 0000 B). Ieșirea din convertorul D/A (50% din diapazon) este comparată cu tensiunea de intrare v_{in} . Dacă tensiunea de intrare este mai mare decât V_{DAC} , atunci bitul rămâne setat, în caz contrar este șters.

Controlul logic al aproximației succesive setează următorul bit cel mai semnificativ (11 0000 0000 B sau 01 0000 0000 B, funcție de rezultatul anterior), iar V_{DAC} este din nou comparat cu v_{in} . Dacă tensiunea de intrare este mai mare decât V_{DAC} , atunci bitul testat rămâne setat; în caz contrar bitul ce a fost testat este șters. Acest proces se repetă până când toți cei zece biți au fost testați; la acest moment rezultatul conversiei este păstrat în registrul cu aproximații succesive. Conversia ia patru cicluri mașină pentru un bit.

Sfârșitul conversiei pe 10 biți este semnalizat de bitul de control **ADCON.4 (ADCI)**. Cei 8 biți superiori ai rezultatului sunt păstrați în registrul de funcții speciale **ADCH**, iar cei doi biți rămași sunt păstrați în **ADCON.7 (ADC1)** și **ADCON.6 (ADC0)**. Utilizatorul poate ignora cei doi biți cei mai puțin semnificativi din **ADCON** și folosi convertorul pentru numai 8 biți (cei 8 biți cei mai semnificativi din **ADCH**). În orice situație, timpul total de conversie este de 50 cicluri mașină. **ADCI** va fi setat, iar flagul de stare **ADCS** va fi resetat 50 cicluri după ce bistabilul de comandă (**ADCS**) este setat.

Biții de control **ADCON.0**, **ADCON.1** și **ADCON.2** sunt folosiți pentru a controla un multiplexor analogic care selectează unul din cele 8 canale analogice. O conversie A / D în desfășurare nu este afectată de un start extern sau software. Rezultatul unei conversii complete rămâne neafectat cu condiția **ADCI = "1"** logic; o nouă conversie A / D, de asemenea în desfășurare, va fi abandonată când se intră în modul mers în gol (idle) sau alimentare redusă (power down). Rezultatul unei conversii complete (**ADCI = "1"** logic) rămâne neafectat când se intră în modul mers în gol (idle).

	7	6	5	4	3	2	1	0
ADCON (C5H)	ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0
	(MSB)							(LSB)

Bit	Simbol	Funcție
ADCON.7	ADC.1	Bitul 1 al rezultatului conversiei A/D
ADCON.6	ADC.0	Bitul 0 al rezultatului conversiei A/D
ADCON.5	ADEX	Activare start extern al conversiei prin STADC 0 = conversia nu poate fi declanșată decât software (prin setarea ADCS) 1 = conversia poate fi declanșată fie software, fie hardware (de un front crescător la STADC)
ADCON.4	ADCI	Flagul de întrerupere al conversiei A/D: acest flag este setat când rezultatul conversiei este gata de a fi citit. O întrerupere poate fi apelată dacă a fost activată. Flagul poate fi șters în rutina de servire a întreruperii. Atât timp cât acest flag este setat, conversia A/D nu poate fi declanșată pentru o nouă conversie. ADCI nu poate fi setat prin software
ADCON.3	ADCS	Start și stare conversie A/D: setarea acestui bit declanșează o conversie A/D. El trebuie setat prin software sau de semnalul extern STADC . Logica convertorului A/D asigură ca acest semnal să fie HIGH , în timp ce convertorul A/D este ocupat. La sfârșitul conversiei, ADCS este resetat în același timp ce flagul de întrerupere este setat. ADCS nu poate fi resetat prin software. O nouă conversie nu poate fi declanșată atât timp cât ADCS sau ADCI sunt HIGH

ADCI	ADCS	STARE CONVERTOR A/D
0	0	Convertorul A/D nu este ocupat; se poate declanșa o conversie
0	1	Convertorul A/D ocupat; startul unei noi conversii este blocat
1	0	Conversia este completă; începerea unei noi conversii este blocată
1	1	Nu este posibil

Dacă ADCI este șters prin software în timp ce ADCS este setat în același timp, o nouă conversie A/D cu același număr al canalului poate fi începută. Dar se recomandă să se reseteze ADCI înainte să se seteze ADCS.

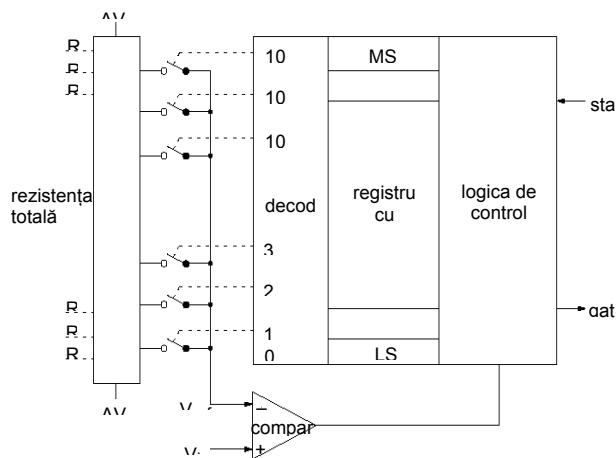
ADCON.2 AADR2 Selecție intrare analogică: adresa codificată binar selectează unul din cei opt
 ADCON.1 AADR1 biți de port analogic din P5 pentru a fi introdus în convertor. Poate fi schimbat
 ADCON.0 AADR0 numai când ADCI și ADCS sunt LOW

AADR2	AADR1	AADR0	CANAL ANALOGIC SELECTAT
0	0	0	ADC.0 (P5.0)
0	0	1	ADC.1 (P5.1)
0	1	0	ADC.2 (P5.2)
0	1	1	ADC.3 (P5.3)
1	0	0	ADC.4 (P5.4)
1	0	1	ADC.5 (P5.5)
1	1	0	ADC.6 (P5.6)
1	1	1	ADC.7 (P5.7)

Registrul de control al convertorului A / D (ADCON)

Rezoluția convertorului A/D și alimentarea analogică.

Figura 6 arată cum este realizat convertorul A/D. Convertorul are proprii săi pini de alimentare (AV_{DD} și AV_{SS}) și doi pini (V_{ref+} și V_{ref-}) conectați la fiecare capăt al scării de rezistențe. Scara are 1023 trepte de lățimi egale, separate printr-o rezistență "R". Prima treaptă este 0.5 × R peste V_{ref-}, iar ultima la 1.5 × R sub V_{ref+}. Aceasta



dă o valoare totală a scării de 1024 × R. Această structură asigură caracteristică monotonă a convertorului A/D și rezultă o cuantizare simetrică a erorii, așa cum se prezintă în figura 5.

Pentru tensiuni de intrare între V_{ref-} și (V_{ref-} + 1/2 LSB, rezultatul de 10 biți al conversiei A/D va fi 00 0000 0000B = 000H. Pentru tensiuni de intrare între V_{ref+} - 3/2 LSB și V_{ref+}, rezultatul conversiei va fi 11 1111 1111B = 3FFH. AV_{ref+} și AV_{ref-} pot fi între AV_{DD} + 0.2V și AV_{SS} - 0.2V. AV_{ref+} trebuie să fie pozitivă în raport cu AV_{ref-}, iar tensiunea de intrare Vin trebuie să fie între

Fig. 6

AV_{ref+} și AV_{ref-}. Dacă tensiunea analogică de intrare este în domeniul de la 2V la 4V, atunci rezoluția de 10 biți poate fi obținută în acest întreg domeniu dacă AV_{ref+} = 4V și AV_{ref-} = 2V.

Rezultatul poate fi calculat întotdeauna cu următoarea formulă:

$$\text{Rezultatul} = 1024 \times \frac{V_{in} - AV_{ref-}}{AV_{ref+} - AV_{ref-}}$$

Aplicație la Convertorul Analog-Numeric

Softul următor își dorește monitorizarea unei tensiuni analogice cuprinse între 11 volți și 14 volți, cu afișarea valorii tensiunii pe unul din biții portului P4. Datorită faptului că microcontrolerul primește tensiune de intrare maximă de 5 volți, pe intrarea analogică, montez un divizor rezistiv prin 3, a.î. la 15 volți voi avea 5 volți; la 12 volți – voi avea la intrarea microcontrolerului 4 volți, ș.a.m.d. Valoarea rezultatului conversiei o voi citi din Regiștrii ADCH și ADCON, dar în ADCON sunt cei 2 biți inferiori – pe care am să-i ignor. Astfel, la 5 volți la intrarea convertorului voi citi în ADCH valoarea 255, la 4 volți am să citesc 204, etc. (datorită faptului că rezultatul convertorului A/D este liniar, SE POATE APLICA REGULA DE TREI SIMPLE, adică, 4 volți la intrare = $(4 \cdot 255) / 5$, ș.a.m.d.).

La 15 volți la măsurare, voi avea (după divizorul rezistiv prin 3) = 5V => 255 (valoare ADCH)

14V => 4,67V => 239

13V => 4,33V => 221

12V => 4V => 204

11V => 3,66V => 187

10V => 3,33V => 170

Datorită faptului că în Asambler este cam dificil de comparat valoarea din ADCH (după ce am citit tensiunea analogică de intrare), (exemplu. dacă valoarea din ADCH = 211, atunci tensiunea analogică de intrare este între 12V și 13V, ca urmare o să aprind Led-ul conectat la pinul de port P4.3), voi apela un artificiu: încarc o valoare de start în registrul R1 pe care tot o compar cu niște valori (187, 204, etc) corespunzătoare valorilor urmărite (11V, 12V, etc.). Dacă valorile nu sunt egale, atunci incrementez valoarea din registrul R1, și iar compar, până se produce până la urmă o coincidență, după care „aprind” LED-ul corespunzător valorii urmărite:

- P4.1 – sub 11V
- P4.2 – 12V
- P4.3 – 13V
- P4.4 – peste 13V

mențiune: flagul Carry este folosit pentru a nu mai iniția altă conversie atâta timp cât deja sunt într-una – ca urmare cât timp are loc o „citire” sunt în bucla:

CONV: jc CONV ;cat timp are loc conversia, stai in bucla asta (c = Carry Flag)

name APLICATIE_CONVERTOR_ANALOG_DIGITAL
\$INCLUDE (reg552.inc)

```

ORG 8000H                ; ADRESA DE INCEPUT
JMP START

ORG 8053H                ; ADRESA INTRERUPERE CONVERTOR
JMP REZULT

ORG 8080H                ;SE SARE PESTE ZONA CU VECTORI DE INTRERUPERE

START:  clr c                ;sterg carry pentru a realiza o conditionare mai tarziu
        MOV P4,#00H          ; STING P4
        mov p5,#00h
        MOV R0,#00H ;SETRG RO
        MOV IEN0,#0C0H      ;ACTIVEZ INTRERUPERILE GLOBALE SI INTR. DUPA CONVERTOR AD.

CONV:   jc CONV              ;cat timp are loc conversia, stai in bucla asta (c = Carry Flag)
        MOV ADCON,#08H       ;PORNESC CONVERTORUL
        setb c                ;am pornit adineaurei Convertorul si nu mai are rost ulterior sa-l mai porneac iar ...
        JMP CONV              ;TREC LA ALTA CONVERSIE

REZULT: MOV R1,#60            ;INCARC VALOAREA DE INCEPUT - alesa arbitrara
        MOV R0,ADCH          ;VAL DIN CON. O DUC IN REG. VAL

IN1:    INC R1
        MOV A,R0
        CJNE A,01H,STOP1     ; AM 11V ? ;IN LOC DE R1 AM SCRIS 01H
        MOV P4,#02H          ;AM 11V , APRINDE P4.1
        MOV ADCON,#00H       ;STERGE FLAGUL DE LA CONERTOR
        clr c                ;sterg caryy, a.i. dupa ce ies din intrerupere, sa activez doar o singura data pornirea convertorului
        RETI                  ;IESIRE INTRERUPERE

STOP1:  CJNE R1,#187,IN1;AM TERMINAT DIAPAZONUL PT. 11V ? , DACA NU, SAR LA IN1 , DACA DA, CONTINUUI!

IN2:    INC R1
        MOV A,R0;
        CJNE A,01H,STOP2; AM 12V ?
        MOV P4,#04H          ;AM 12V , APRINDE P4.2
        MOV ADCON,#00H ;STERGE FLAGUL DE LA CONERTOR
        clr c
        RETI                  ;IESIRE INTRERUPERE

STOP2:  CJNE R1,#204,IN2 ;AM TERMINAT DIAPAZONUL PT. 12V ? , DACA NU, SAR LA IN2 , DACA DA, CONTINUUI!

IN3:    INC R1
        MOV A,R0
        CJNE A,01H,STOP3; AM 13V ?
        MOV P4,#08H          ;AM 13V , APRINDE P4.3
        MOV ADCON,#00H ;STERGE FLAGUL DE LA CONERTOR
        clr c
        RETI                  ;IESIRE INTRERUPERE

STOP3:  CJNE R1,#221,IN3 ;AM TERMINAT DIAPAZONUL PT. 11V ? , DACA NU, SAR LA IN1 , DACA DA, CONTINUUI!

        MOV P4,#10H          ;AM DEPASIRE DE 13V
        MOV ADCON,#00H ;STERGE FLAGUL PENTRU O NOUA INTRERUPERE
        clr c
        RETI

END
    
```