PROIECT

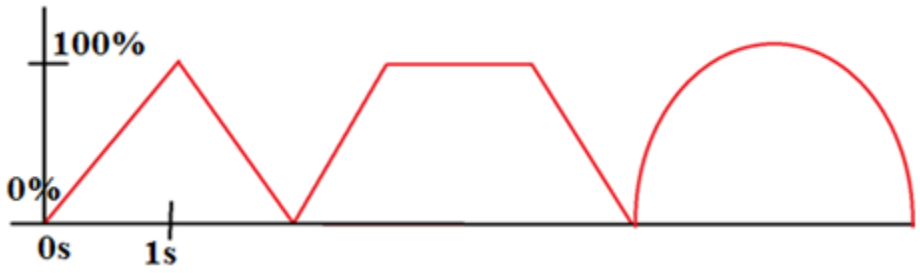
MICROPROCESOARE SI MICROCONTROLERE

PWM VARIABIL-PROIECT 2

Student:Lambru Eusebiu-Vasilica

Grupa:5302

1.Enunt: Sa se genereze un sistem cu uC PIC care sa controleze prin metoda PWM un motor de curent continuu astfel incat tensiunea echivalenta sa varieze conform cu figura:



Observatie: Ca sa ajunga la turatie maxima, trebuie un timp de cateva secunde(1sec/2sec)

PWM (Pulse Width Modulation - modularea factorului de umplere al unui semnal) permite controlul circuitelor analogice din domeniul digital. Un controller PWM este in esenta un convertor digital-analog (DAC) pe un singur bit. Această modulație vine în sprijinul multor aplicații deoarece înseamnă consum mic, eliminarea zgomotelor și aplicații cu cost redus.

Puterea aplicata pe motor poate fi controlata prin variatia factorului de umplere al pulsurilor si totodata, variaza tensiunea aplicata pe terminalele motorului. Prin schimbarea/modularea pulsurilor , viteza motorului poate fi controlata, de exemplu, cu cat sta mai mult un puls in “ON” , cu cat creste mai mult viteza de rotatie a motorului si invers, cu cat sta mai putin in “ON” , cu atat motorul se roteste mai incet.

2.Referinte cod:

Pentru triunghi si trapez am calculat panta crescatoare/descrescatoare in 10 puncte(N=10). Deoarece motorul nu ajunge instantaneu la turatie maxima, am ales un timp de crestere de 1 sec.

Cum numarul de tranzitii il stim, N=10(step_PWM=10%) si timpul de crestere Tc=1sec(PWM 0%..step..100%), putem calcula perioada: T = Tc/N = 0.1s = 100ms. De aici deducem ca 1 punct de cuantizare corespunde a 100ms delay. De asemenea, stim ca pentru 1 punct avem nevoie de 10 stari

(i stari in “1” logic si N-i stari in "0" logic). Rezulta ca pentru fiecare din cele 10 stari ale unui punct ii corespunde o intarziere de 10ms(de accea am folosit o bucla de temporizare in functia delay_10ms).

Similar , la trapez, am folosit acceasi metoda ca la triunghi cu deosebirea ca sta un timp mai indelungat in punctul maxim(1 secunda) in care semnalul sta in "1" logic .

La sinusoida , am ales un timp de 4 secunde, si cum avem 10 puncte pe panta cresc/descresc. rezulta ca vom sta cate 400ms in fiecare punct al sinusului. Avand in vedere ca punctele sinusului sunt numere reale, am inmultit formula de generare a punctelor cu 32. Avand o perioada de 10ms la fiecare tranzitie, perioada ne da aproximativ 400ms.

Formula de generare a sinusului: $32*\sin(n*\pi/20)$,n=1,2,3...20

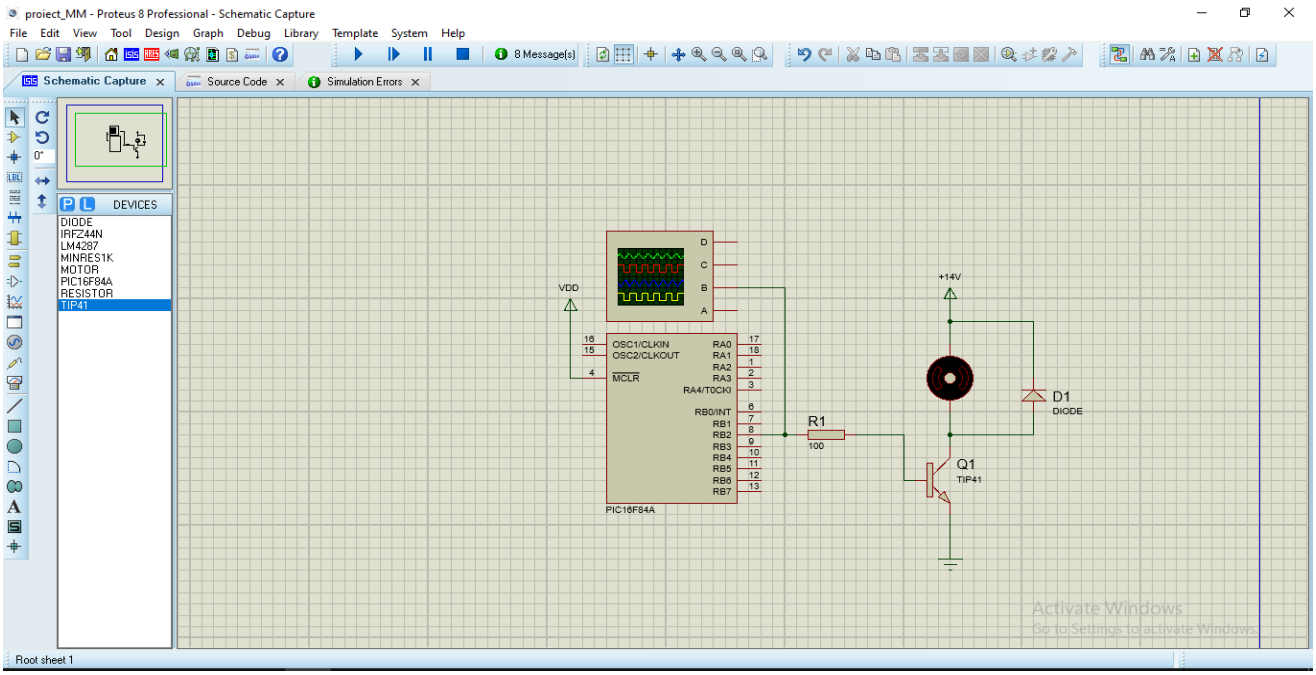
factorii de umplere rezultati dupa formula de generare:
5,10,15,19,23,26,28,30,31,32,31,30,28,26,23,19,15,10,5,0

Pentru a genera forma de unda, vom folosi o subrutina "puncte_sin" si o vom apela atunci cand vrem sa incarcam o valoare pentru a genera punctele sinusului.

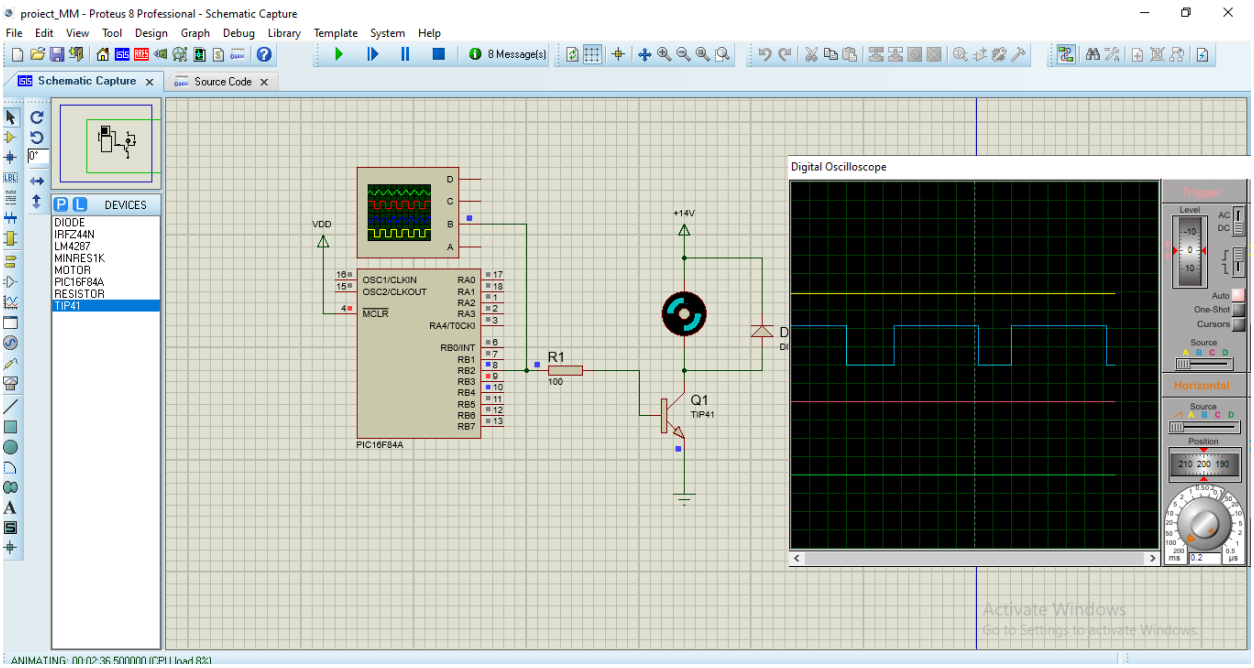
Subrutina puncte_sin va returna constanta data de PCL(program counter low) in care vom adauga prin numarul de ordine al punctului intr-un registru n, in care mai apoi, il vom muta intr-un registru p pentru a genera delay-ul.

Pentru implementare, am ales microcontroler-ul PIC16F84A si frecventa de oscilatie de 4MHZ pentru ca 1cm(o instructiune simpla) = 1us . Ca si port am folosit PORTB, mai exact pinul 2(RB2).

3.Schema circuit in proteus:



Folosim Digital Oscilloscope pentru a vizualiza forma de unda:



Componente:

- R=100ohm
- Q1(npn) ->l-am folosit ca si switch pentru a transfera PWM-ul la motor
- Dioda D1
- sursa de tensiune aplicat la motor de +14V
- DC motor
- microcontroler PIC16F84A

4.COD ASM

```
#include p16f84a.inc
i       equ 0x20        ;i=0...N pentru triunghi/trapez
N       equ 0x21        ;Nr de tranzitii
x       equ 0x22        ;folosit pentru generarea factorului de umplere
j       equ 0x23        ;il folosim in bucla pentru generare delay
k       equ 0x24        ;il folosim pentru calcularea delay-ului de 10ms
p       equ 0x25    ;p retine factorul de umplere din subrutina puncte_sin
pct equ d'32'     ;punctul maxim al sinusului la 90 grade
n       equ 0x26    ;n reprezinta factorul de umplere al sinusului
main:
BCF STATUS,RP1
BSF STATUS,RP0          ;{01}-bank1
MOVLW B'11111011'     ;lasam activ doar RB2
MOVWF TRISB
BCF STATUS,RP1
BCF STATUS,RP0          ;{00}- BANK0

cresc_tri:
MOVLW D'10'
MOVWF N         ;N=10 , nr de tranzitii pe panta cresc/descresc
CLRF i    ;stergem i-ul pentru a incepe iteratia
cresc_tri_1:       ;stam i apeluri in "1"

MOVF i,0          ;acc=i, actualizare Z
BTFSC STATUS,Z;testez bitul Z
GOTO cresc_tri_0 ;Z==0, skip GOTO
```

```
MOVWF x          ;x=w(acc)=i=>x=i
BSF PORTB,2
;x=acc=i
CALL delay_x

cresc_tri_0:       ;stam N-i apeluri in "0"
MOVF i,0
SUBWF N,0        ;acc=N-i
BTFSC STATUS,Z
GOTO desc_tri  ;i==N, Z=1
MOVWF x                    ;x=N-i
BCF PORTB,2        ;i~=N, Z=0
CALL delay_x
INCF i,1 ;i++ => creste numarul de apeluri in 1
GOTO cresc_tri_1


desc_tri:           ;este la fel ca la panta crescatoare
;doar ca generat in sens invers
clrf i      ;stergem i-ul pentru a incepe panta descrescatoare
desc_tri_0:                ;i apeluri in 0
MOVF i,0          ;acc=i, actualizare Z
BTFSC STATUS,Z;Z==0, skip goto
GOTO desc_tri_1 ;i==0, Z=1 =>N-i=N-0 apeluri in "1"
MOVWF x                    ;x=acc=i
BCF PORTB,2
CALL delay_x

desc_tri_1:        ;N-i apeluuri in 1
```

```
MOVF i,0          ;i=w(acc)
SUBWF N,0         ;acc=N-i
;stam in bucla de N-i ori
BTFSC STATUS,Z
GOTO trapez_cresc  ;i==N, Z=1, sari la trapez
MOVWF x                    ;x->w->N-i
BSF PORTB,2        ;i~=N, Z=0
CALL delay_x      ;producem delay-ul de x ori
INCF i,1 ;incrementam i pana la N
GOTO desc_tri_0

;GENERARE TRAPEZ-la fel cu triunghiul, diferenta fiind ca sta
;un timp mai indelungat (1s) in punctul maxim
;intarzierea se face pe portul RB2
trapez_cresc:
BCF PORTB,2
CALL delay_10ms           ;genereaza un delay de 10ms intre triunghi si trapez
clrf i
cresc_trap_1      ;i apeluri in 1 logic
MOVF i,0          ;acc=i, actualizare Z
BTFSC STATUS,Z
GOTO cresc_trap_0 ;i==0, Z=1
MOVWF x                    ;x=acc=i
BSF PORTB,2       ;i~=0, Z=0
CALL delay_x
cresc_trap_0      ;N-i apeluri in 0 logic
MOVF i,0
SUBWF N,0         ;acc=N-i
BTFSC STATUS,Z
```

```
GOTO platou_trapez  ;i==N, Z=1
MOVWF x                     ;x=N-i
BCF PORTB,2        ;i~=N, Z=0
CALL delay_x

INCF i,1
GOTO cresc_trap_1



; Deosebirea fata de triunghi consta ca "punem"
; semnalul in 1 logic timp de 1s in punctul maxim
platou_trapez:
movlw d'100'
movwf x            ;x<-100
bsf PORTB,2
CALL delay_x     ;generam 100*10ms=1s delay
bcf PORTB,2
CALL delay_10ms ; lasam un delay de 10ms inainte de panta descresc.

;descrescator
;panta descrescatoare a trapezului
desc_trap:
clrf i      ;stergem i sa nu ramana valoarea veche
desc_trap_0      ;i apeluri in 0 logic
MOVF i,0           ;acc=i, actualizare Z
BTFSC STATUS,Z
GOTO desc_trap_1 ;i==0, Z=1
MOVWF x                     ;x=acc=i
BCF PORTB,2      ;i~=0, Z=0
```

```
CALL delay_x



desc_trap_1                 ;N-i apeluri in 1 logic
MOVF i,0
SUBWF N,0       ;acc=N-i
BTFSC STATUS,Z
goto loop_sin  ;i==N, Z=1 , genereaza panta crescatoare, sin
MOVWF x                     ;x=N-i
BSF PORTB,2       ;i~=N, Z=0
CALL delay_x
INCF i,1
GOTO desc_trap_0

;generare sin

;generare sinus, panta crescatoare



loop_sin:
BCF PORTB,2
CALL delay_10ms           ;genereaza un delay de 10ms intre trapez si sin
CLRF i    ;stergem i-ul pentru a incepe iteratia
MOVLW D'0'
MOVWF n ;n il punem in 0 astfel ca la primul apel
;al functiei puncte_sin, PCL+n sa ia prima valoare
loop_sin_1:       ;genereaza p apeluri in "1" logic
INCF n,1;atunci cand ajunge in punctul maxim 32
```

```
;ne reintoarcem in loop_sin_1 si pt a nu genera de 2 ori
;in acelasi punct,incrementam n-ul de la inceput
MOVF n,0
CALL puncte_sin             ;la intoarcerea din apel, va avea
;incarcat in acumulator valoarea n corespunzatoare punctelor
;sinusului
MOVWF p         ;din acc, salvam val. in p, pentru ca reprezinta
;factorul de umplere de pe panta cresc.

MOVF p,0 ;acc<-acc
BTFSC STATUS,Z
GOTO loop_sin_0 ;i==0, Z=1
MOVWF x                     ;
BSF PORTB,2                 ;i~=0, Z=0
CALL delay_x
loop_sin_0:
MOVF p,0
SUBLW pct       ;vom genera pct-p apeluri in 0 logic
BTFSC STATUS,Z
GOTO loop_sin_1  ;i==N, Z=1
MOVWF x
BCF PORTB,2       ;i~=N, Z=0
CALL delay_x

MOVF p,0
BTFSC STATUS,Z;daca p=0=> a ajuns la final
GOTO cresc_tri  ;dupa ce am terminat generarea sinusului
;ne intoarcem la inceput
GOTO loop_sin_1
```

```
;{0,5,10,15,19,23,26,28,30,31,32,31,30,28,26, 23,19,15,10,5,0};
;PCL - program counter pe octetul de low

puncte_sin: ; aici pastram pct. sinusului pentru o accesare
;facila.
addwf PCL
retlw d'0'          ;am luat 21 de puncte deoarece n-ul se incrementeaza la inceput
retlw d'5'
retlw d'10'
retlw d'15'
retlw d'19'
retlw d'23'
retlw d'26'
retlw d'28'
retlw d'30'
retlw d'31'
retlw d'32'
retlw d'31'
retlw d'30'
retlw d'28'
retlw d'26'
retlw d'23'
retlw d'19'
retlw d'15'
retlw d'10'
retlw d'5'
retlw d'0'
```

```
delay_x:
CALL delay_10ms
DECFSZ x,1        ;(8+(5k+4)j+2+1+2)x-1+2=(13+(5k+4)j)x+1
GOTO delay_x
RETURN



delay_10ms:
movlw d'1'                ;init(j)=2cm
movwf j

loop_k:                   ;init(k)=2cm
movlw d'99'
movwf k

Loop_10ms:
NOP               ;2nop=2cm
NOP
decfsz k,1        ;1(2)cm-> 2cm cand k=0 =>skip GOTO
;(2+1+2)k-1+2+1+2)j=(5k+4)j
;BSF+CALL+init(k)+init(j)+delay_10ms+return=1+2+2+2+(5k+4)j+2-1=
;8+(5k+4)j=8+(5*199+4)*10=9998cm aprox 10ms
goto Loop_10ms          ;2cm
decfsz j,1        ;1(2)cm
```

goto loop_k

RETURN ;+2cm =>9998 + 2cm =>10ms

NOP

end

5.CALCUL DELAY:

Tc=1sec(PWM 0%..step..100%)T = Tc/N = 0.1s = 100ms

fosc=4MHz $4 \cdot 10^6$ tacte sec. => $1 \cdot 10^6$c.m./sec => 1c.m.(1instr.simplă)=1us

BSF + CALL + init(k) + init(j) + delay_10ms + return =1 + 2 + 2 + 2 + (5k+4)j + 2-1 = 8+(5k+4)j = 9998cm aprox 10ms

Calcul delay-ului in functia delay_x:

;(8+(5k+4)j+2+1+2)x-1+2=(13+(5k+4)j)x+1= 9998cm * x + 1

Exemplu : generare panta crescatoare, i=1 => x=i apeluri in "1" si x=N-i apeluri in "0"

Pentru "1" : i=1 =>x=1 =>delay_x aprox. 10ms * x + 1 => delay_x are 10ms delay

Pentru "0":i=1 , N=10 =>x=N-i=10-1=9 =>delay_x = 10ms*(N-i)+1=10ms*9 + 1 =90ms =>T1+T0=100ms =T

COD C :

```
#include <htc.h>

#define _XTAL_FREQ 4000000


unsigned char i,N,x,pct=32,nr=20,pct_maxim=100;

//signed char i;

const unsigned int table[50] =
```

```
{5,10,15,19,23,26,28,30,31,32,31,30,28,26, 23,19,15,10,5,0};

void main(void)
{       TRISB = 0b11100011;

while(1)
{       //triunghi cresc
N=10;
for (i=0; i<=N; i++)
{       //for(x=1; x<=i; x++)
for(x=i; x>0; x--)
{ RB2=1; __delay_ms(10); }
for(x=N-i; x>0; x--)
{ RB2=0;__delay_ms(10); }
}
//triunghi descrescator
for (i=1; i<=N-1; i++)
{       //for(x=1; x<=i; x++)
for(x=N-i; x>0; x--)
{ RB2=1; __delay_ms(10); }
for(x=i; x>0; x--)
{ RB2=0; __delay_ms(10); }
}

//----------forma trapez------
//TRAPEZ CRESC.
for (i=0; i<=N; i++)
{       //for(x=1; x<=i; x++)
for(x=i; x>0; x--)
```

```
{ RB2=1; __delay_ms(10); }
for(x=N-i; x>0; x--)
{ RB2=0; __delay_ms(10); }
}
//forma trapez platou
for(x=pct_maxim; x>0; x--){
{ RB2=1; __delay_ms(10); }        //sta 100*10ms=1s in 1
}
//forma trapez descrescator
for (i=1; i<=N-1; i++)
{          //for(x=1; x<=i; x++)
for(x=N-i; x>0; x--)
{ RB2=1; __delay_ms(10); }
for(x=i; x>0; x--)
{ RB2=0;__delay_ms(10);}
}


//x=factor de umplere
//N=nr de puncte pe panta cresc/descresc
//pct =32, pct repre valoarea maxima a sinusului ;a 90grade
//sta in 1 logic p iteratii , 0 logic pct - p iteratii

//generare sinus
for (i=0; i<nr; i++)
{
for(x=table[i]; x>0; x--)//table[i] apeluri in "1" logic
{ RB2=1; __delay_ms(10); }
for(x=pct-table[i]; x>0; x--)
```

{

RB2=0; __delay_ms(10); } //pct-table[i] apeluri in "0" logic

}

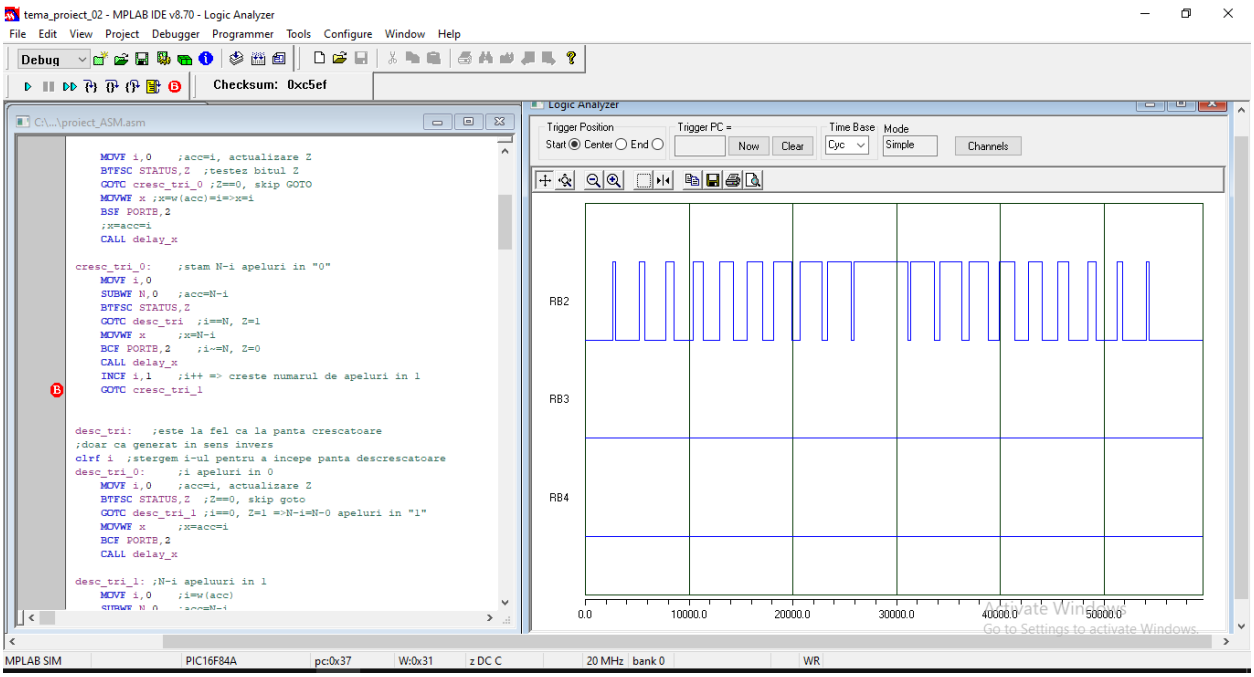
}

}

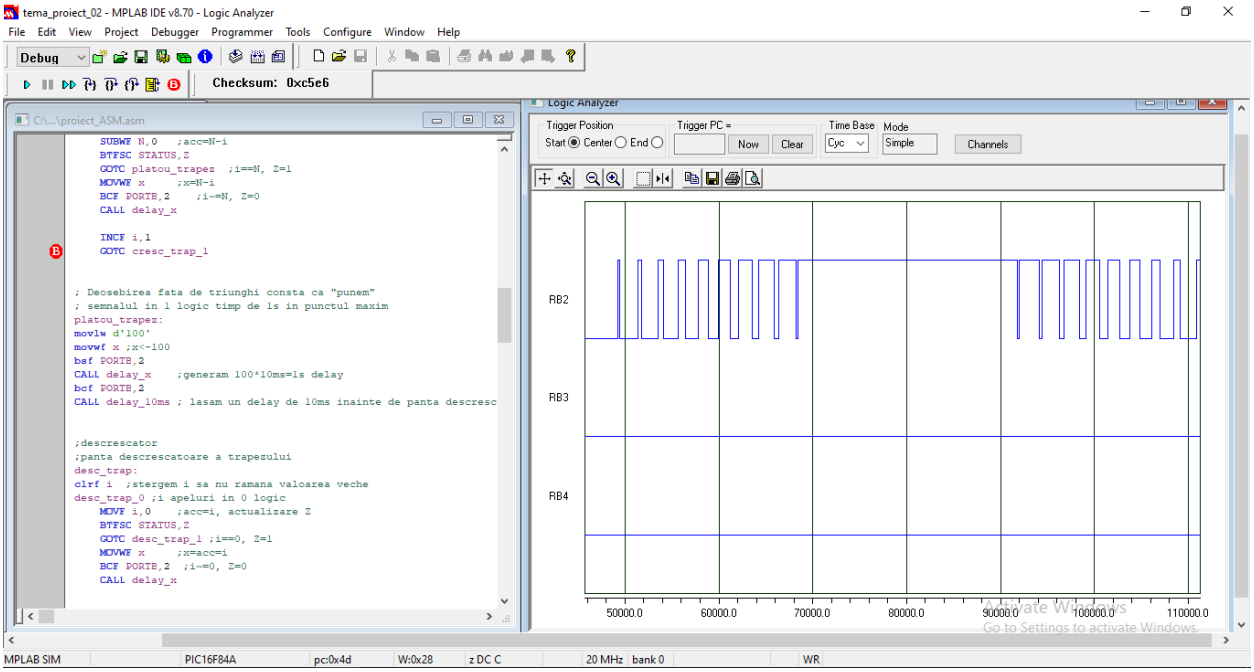
Forma de unda cod ASM:

Am folosit un delay mai mic pentru a vizualiza intreaga forma de unda.

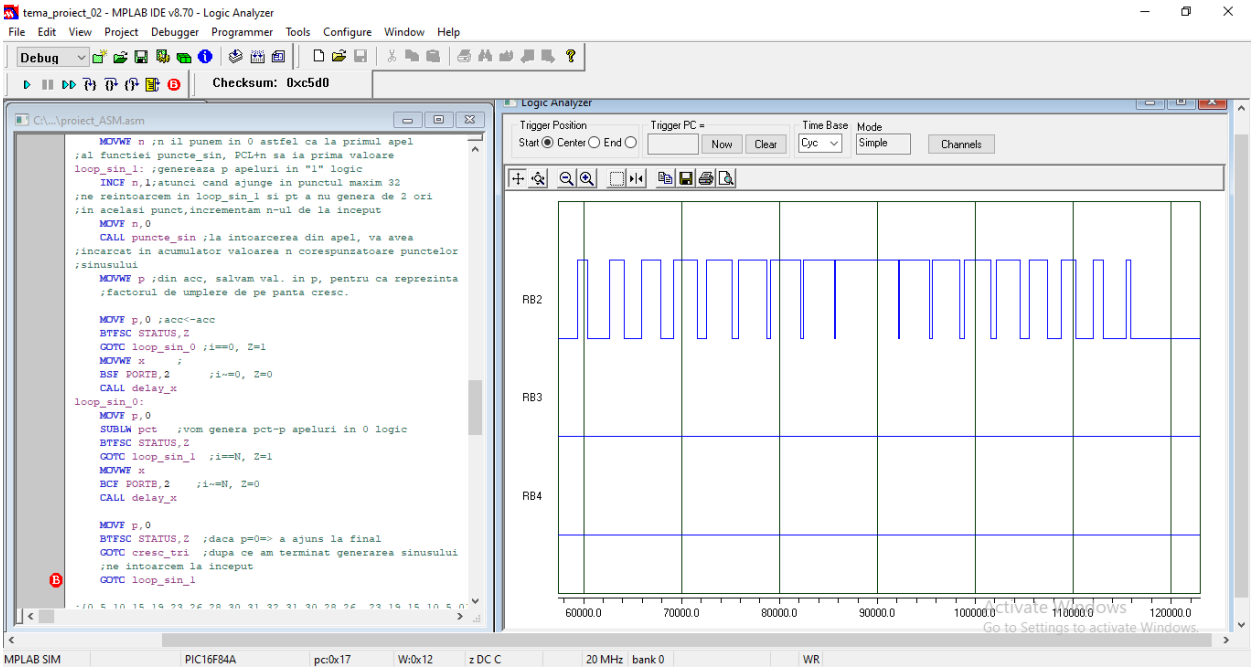
Triunghi:



Trapez:


tema_proiect_02 - MPLAB IDE v8.70 - Logic Analyzer
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug
Checksum: 0xc5e6
C:\...\proiect_ASM.asm
SUBWF N,0 ;acc=N-i
BTFSC STATUS,Z
GOTC platou_trapez ;i==N, Z=1
MOVWF x ;x=N-i
BCF PORTB,2 ;i~=N, Z=0
CALL delay_x
INCF i,1
GOTC cresc_trap_1
; Deosebirea fata de triunghi consta ca "punem"
; semnalul in 1 logic timp de 1s in punctul maxim
platou_trapez:
movlw d'100'
movwf x ;x<-100
bsf PORTB,2
CALL delay_x ;generam 100*10ms=1s delay
bcf PORTB,2
CALL delay_10ms ; lasam un delay de 10ms inainte de panta descresc
;descrescator
;panta descrescatoare a trapezului
desc_trap:
clrf i ;stergem i sa nu ramana valoarea veche
desc_trap_0 ;i apeluri in 0 logic
MOVF i,0 ;acc=i, actualizare Z
BTFSC STATUS,Z
GOTC desc_trap_1 ;i==0, Z=1
MOVWF x ;x=acc=i
BCF PORTB,2 ;i~=0, Z=0
CALL delay_x
Logic Analyzer
Trigger Position
Start Center End
Trigger PC =
Now Clear
Time Base
Cyc
Mode
Simple
Channels
RB2
RB3
RB4
50000.0
60000.0
70000.0
80000.0
90000.0
100000.0
110000.0
Activate Windows
Go to Settings to activate Windows.
MPLAB SIM
PIC16F84A
pc:0x4d
W:0x28
z DC C
20 MHz
bank 0
WR

Sinus:


tema_proiect_02 - MPLAB IDE v8.70 - Logic Analyzer
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug
Checksum: 0xc5d0
C:\...\proiect_ASM.asm
MOVWF n ;n il punem in 0 astfel ca la primul apel
;al functiei puncte_sin, PCL+n sa ia prima valoare
loop_sin_1: ;genereaza p apeluri in "1" logic
INCF n,1;atunci cand ajunge in punctul maxim 32
;ne reintoarcem in loop_sin_1 si pt a nu genera de 2 ori
;in acelasi punct,incrementam n-ul de la inceput
MOVF n,0
CALL puncte_sin ;la intoarcerea din apel, va avea
;incarcat in acumulator valoarea n corespunzatoare punctelor
;sinusului
MOVWF p ;din acc, salvam val. in p, pentru ca reprezinta
;factorul de umplere de pe panta cresc.
MOVF p,0 ;acc<-acc
BTFSC STATUS,Z
GOTC loop_sin_0 ;i==0, Z=1
MOVWF x ;
BSF PORTB,2 ;i~=0, Z=0
CALL delay_x
loop_sin_0:
MOVF p,0
SUBLW pct ;vom genera pct-p apeluri in 0 logic
BTFSC STATUS,Z
GOTC loop_sin_1 ;i==N, Z=1
MOVWF x
BCF PORTB,2 ;i~=N, Z=0
CALL delay_x
MOVF p,0
BTFSC STATUS,Z ;daca p=0=> a ajuns la final
GOTC cresc_tri ;dupa ce am terminat generarea sinusului
;ne intoarcem la inceput
GOTC loop_sin_1
Logic Analyzer
Trigger Position
Start Center End
Trigger PC =
Now Clear
Time Base
Cyc
Mode
Simple
Channels
RB2
RB3
RB4
60000.0
70000.0
80000.0
90000.0
100000.0
110000.0
120000.0
Activate Windows
Go to Settings to activate Windows.
MPLAB SIM
PIC16F84A
pc:0x17
W:0x12
z DC C
20 MHz
bank 0
WR