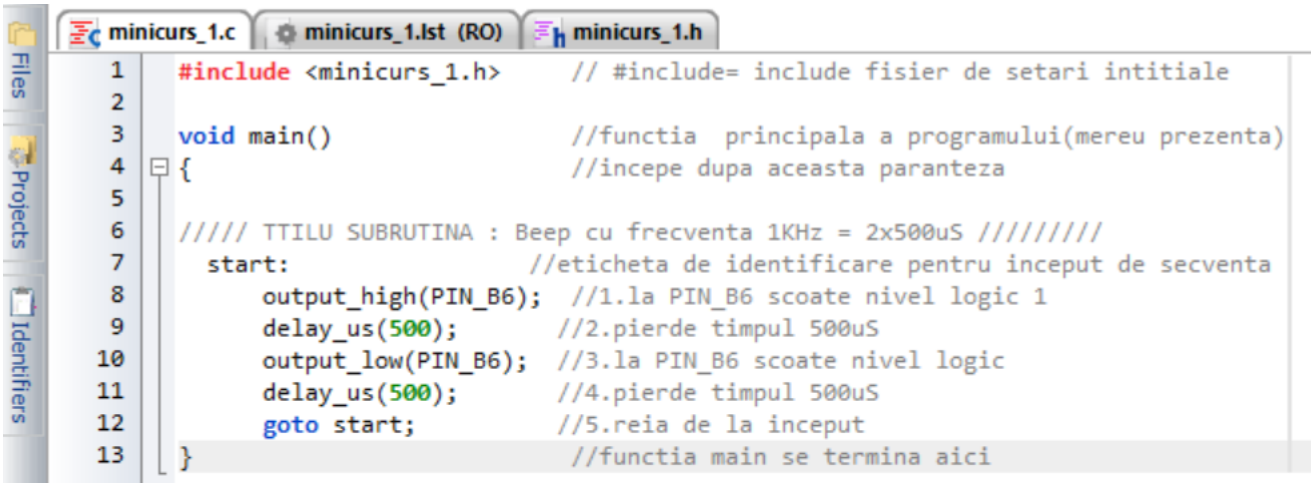


### 3.Instructiuni de iesire (continua)

#### 3.1 Beeper/Instructiunea For (.....)/ Numere intregi pe 8 biti: **int8**

In aplicatia anterioara *buzzer-ul* suna continuu. Acum il vom face sa sune pe o durata anume. Avem programul deja in CCSC:



```

1  #include <minicurs_1.h> // #include= include fisier de setari initiale
2
3  void main() //functia principala a programului(mereu prezenta)
4  { //incepe dupa aceasta paranteza
5
6  //TITLU SUBRUTINA : Beep cu frecventa 1KHz = 2x500uS //TITLU SUBRUTINA
7  start: //eticheta de identificare pentru inceput de secventa
8  output_high(PIN_B6); //1.la PIN_B6 scoate nivel logic 1
9  delay_us(500); //2.pierde timpul 500uS
10 output_low(PIN_B6); //3.la PIN_B6 scoate nivel logic
11 delay_us(500); //4.pierde timpul 500uS
12 goto start; //5.reia de la inceput
13 } //functia main se termina aici

```

Trebuie sa facem in asa fel incit rutina de mai sus sa ruleze doar un timp.

Nu putem utiliza instructiunea **delay**.

La fiecare rulare a rutinei se va trimite o perioada (oscilatie) catre buzzer(1 /pauza/0/pauza).

Durata unei perioade este de 1mS (f=1KHz).

Daca rulam rutina de mai multe ori atunci sunetul se va auzi o durata limitata de timp .

Cit dorim sa se auda sunetul? Sa incepem cu 200mS (0,2S).

Va fi un beep scurt dar ulterior vom creste durata.

Va trebui sa reluam programul (de unde apare **goto start**) de 200 de ori.

Aceasta se face utilizind un contor care numara de cite ori a rulat rutina.

Contorul este de fapt o variabila, adica valoarea sa se va modifica in timpul rularii. In cazul nostru se va modifica de la 1 la 200. Aceasta variabila se poate codifica pe 8 biti in sistemul binar deoarece numarul maxim ce se poate forma cu 8 biti este  $2^8=256$ , iar  $200 < 256$ . De fapt in binar se numara de la 0 la 255 si sint exact 256 de numere. Aceasta variabila va ocupa o singura locatie in memoria RAM a PIC-ului.

Variabila va fi de tip intreg pe 8 biti si va fi definita inainte de utilizare.

**Int8 contor;**

Pentru programare vom utiliza instructiunea **for**.

**for(contor=1;contor<200;contor=contor+1)**

{executa ce scrie aici}

Acoladele se aduga daca sint mai multe instructiuni de executat. Daca se va executa o singura instructiune atunci nu se mai scriu acoladele.

In program gasiti descrierea actiunilor la executia instructiunii **for(...)**

```

1  #include <minicurs_1.h>    // #include= include fisier de setari initiale
2
3  int8 contor;             //aici definim variabila contor ca fiind de tip intreg ...
4                          //...pe 8 biti
5
6  void main()              //functia principala a programului(mereu prezenta)
7  {                        //incepe dupa aceasta paranteza
8
9      ///// TTILU SUBRUTINA : Beep cu frecventa 1KHz = 2x500uS ///////////
10     //// durata sunetului 200mS ///////////////////////////////////////////
11
12     for(contor=1;contor<200;contor=contor+1)
13         //incepe cu contor=1
14         //verifica daca contor<200
15         //daca e adevarat atunci contor se maresta cu 1 si executa 1...4
16         //daca contor>200 iese din rutina
17     {
18         output_high(PIN_B6); //1.la PIN_B6 scoate nivel logic 1
19         delay_us(500);       //2.pierde timpul 500uS
20         output_low(PIN_B6);  //3.la PIN_B6 scoate nivel logic
21         delay_us(500);       //4.pierde timpul 500uS
22     }
23
24 }                          //functia main se termina aici

```

Asa cum este definita variabila contor aceasta va putea avea valoarea maxima 255.

Daca dorim sa avem un sunet mai lung va trebui sa definim variabila pe 16 biti:

**Int 16 contor;**

In acest fel, variabila contor poate avea valoarea maxima  $2^{16}-1=65535$  (atentie este intre 0 si 65535).

Ca atare instructiunea va fi:

**for(contor=1;contor<1000;contor=contor+1) // va rula pentru 1000 cicli = 1000mS=1S**  
 {executa ce scrie aici}

O variabila se poate initializa (poate avea o valoare inca de la definirea sa):

**Int8 contor=3;** // contor va avea valoarea 3

La toate programele din aceasta aplicatie se poate inlocui secventa:

**Output\_high(PIN\_B6);**

**Delay\_us(500);**

**Output\_low(PIN\_B6);**

**Delay\_us(500);**

cu instructiunea **output\_toggle**.

Aceasta face ca iesirea sa isi schimbe valoarea din 0 in 1 logic si invers la fiecare trecere.

```

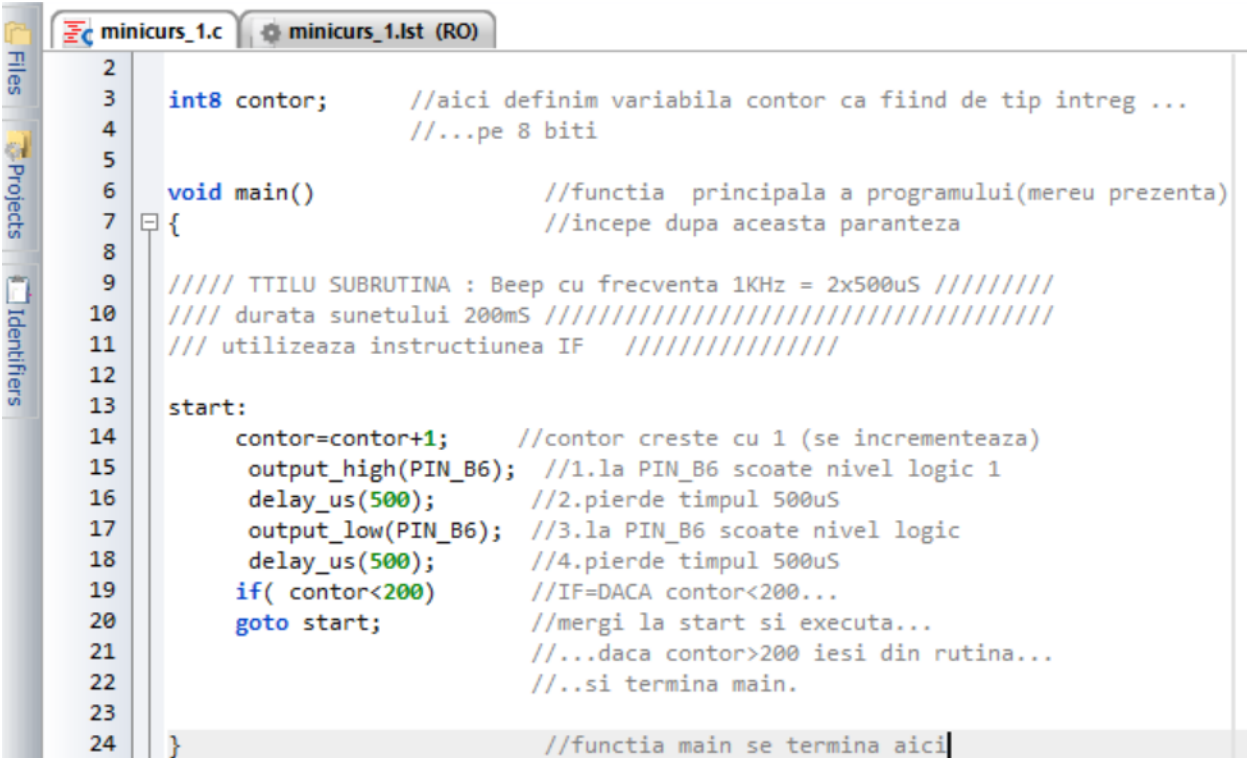
{
output_toggle(PIN_B6);
delay_us(500);
}

```

### 3.2 Beeper / Instructiunea IF

Exista mai multe solutii pentru ca *beeper-ul* sa sune o durata anume. Acum vom folosi instructiunea **IF**.

Instructiunea **IF** functioneaza ca in exemplul urmator:



```
2
3 int8 contor; //aici definim variabila contor ca fiind de tip intreg ...
4 //...pe 8 biti
5
6 void main() //functia principala a programului(mereu prezenta)
7 { //incepe dupa aceasta paranteza
8
9 //TITLU SUBRUTINA : Beep cu frecventa 1KHz = 2x500uS //
10 //durata sunetului 200ms //
11 //utilizeaza instructiunea IF //
12
13 start:
14 contor=contor+1; //contor creste cu 1 (se incrementeaza)
15 output_high(PIN_B6); //1.la PIN_B6 scoate nivel logic 1
16 delay_us(500); //2.pierde timpul 500uS
17 output_low(PIN_B6); //3.la PIN_B6 scoate nivel logic
18 delay_us(500); //4.pierde timpul 500uS
19 if( contor<200) //IF=DACA contor<200...
20 goto start; //mergi la start si executa...
21 //...daca contor>200 iesi din rutina...
22 //..si termina main.
23
24 } //functia main se termina aici|
```

Am introdus iar eticheta **start** si am adaugat instructiunea **IF** la sfirsit. Instructiunea se putea pune si la inceput dupa incrementarea contorului.

Pentru a fi rigurosi in generarea unei unde cu perioade de 1 si 0 egale trebuie sa tinem cont ca saltul la inceput va dura ceva timp si atunci ultima intirziere va trebui sa fie mai mica de 500uS.

Ca sa stiti cu cit sa modificati trebuie sa listati programul in versiunea de asamblare (meniu VIEW/CASM LIST). Numarati instructiunile de la compararea contorului cu 200 si saltul la inceput (start).

Tineti cont ca la frecventa de 8MHz durata de executie a unei instructiuni este de 0,5uS. Scadeti durata intirzierii delay pentru iesire in 0 logic corespunzator.

Nu e usor sa fii riguros! Din fericire pentru inceput nici nu trebuie!

Pina acum probabil nu ati facut greseli sau nu ati avut nevoie sa stiti perfect mnemonica instructiunilor. Dar ce se intimpla cind gresiti sau nu stiti sa scrieti o instructiune?

Puteti utiliza sistemul de ajutor (*help*). Apasati F1 si cautati instructiunea dorita. Uneori este suficient sa scrieti ce va amintiti din instructiune si apoi sa apasati F1.

Daca apar erori dupa compilare cititi in partea de jos a ecranului linia unde se afla eroarea. Linia este marcata pe marginea din stanga a programului.

Uneori nu se afiseaza linia unde este eroarea ci se afiseaza ultima linie a programului. Cititi cu atentie eroarea si vedeti unde ati scris ultima oara ceva.

Modificari:

- modificati frecventa sunetului generat
- modificati programul pentru a genera trei sunete de durate si frecvente diferite
- modificati programul pentru a genera trei sunete de durate si frecvente diferite apoi o pauza de 1 secunda apoi reia de la inceput

Dupa ce studiat aplicatia urmatoare (despre intrari):

- generati un sunet cu o anumita frecventa atunci cind apasati un buton (puteti genera 4 frecvente diferite utilizind butoanele disponibile)
- aprindeti LED B7 cind apasati un buton si apoi generati sunetul specific