

4.1 Instructiuni de intrare /INPUT

Pentru a citi o intrare folosim instructiunea **input**.

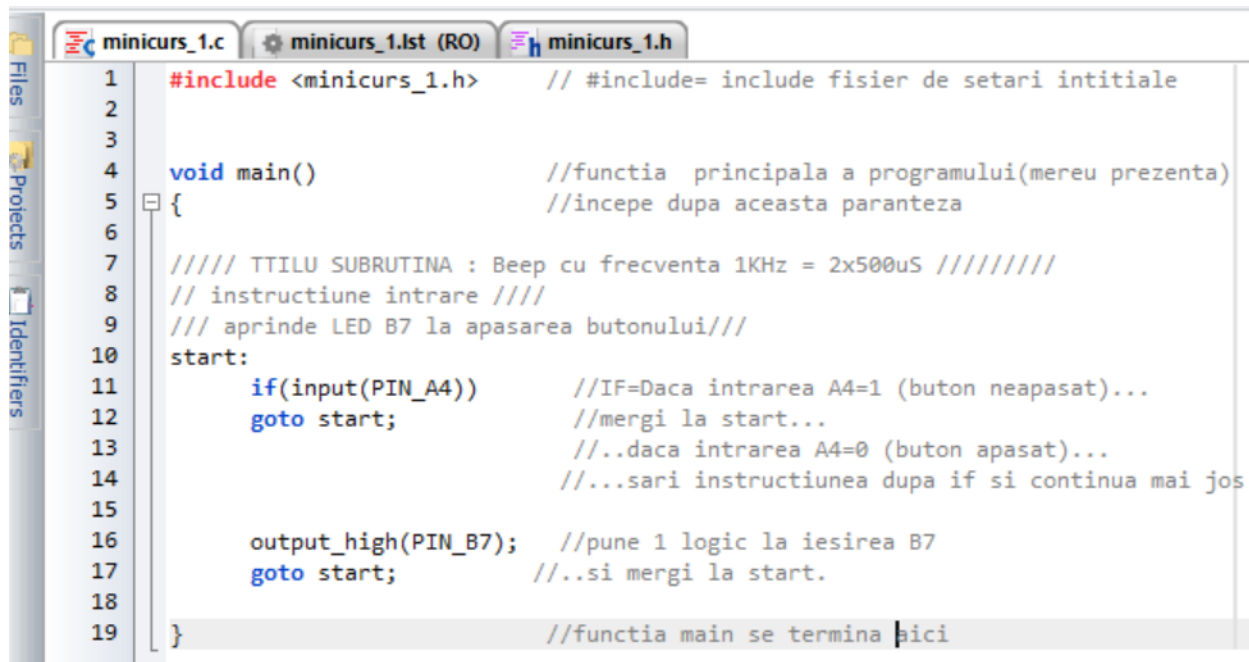
```
input(PIN_A4) // citeste intrarea A4 si verifica daca are valoarea 1 logic;
```

```
linput(PIN_A4) // citeste intrarea A4 si verifica daca are valoarea 0 logic;
```

Remarcati semnul exclamarii inainte de *input* la verificarea cu zero.

Urmatorul exemplu arata cum se poate folosi instructiunea **input** pentru a citi o intrare:

Programul incepe cu citirea intrarii A4. Atita timp cit nu se apasa pe buton, intrarea A4 ramine pe 1 logic si programul va executa instructiunea ce urmeaza imediat dupa **if (goto start)** si va executa de la **start**. Asta inseamna ca va cicla intre cele doua instructiuni **if** si **goto** pina ce butonul se va apasa si intrarea va fi 0 logic. In acea situatie programul va sari instructiunea **goto start** si va executa instructiunea urmatoare pe care o cunoastem deja (**output**). Aceasta va pune un 1 logic la iesirea B7 si astfel se va aprinde LED-ul LB7.



```

1  #include <minicurs_1.h> // #include= include fisier de setari initiale
2
3
4  void main() //functia principala a programului(mereu prezenta)
5  { //incepe dupa aceasta paranteza
6
7  ////// TTILU SUBRUTINA : Beep cu frecventa 1KHz = 2x500uS ///////////////
8  // instructiune intrare ////
9  // aprinde LED B7 la apasarea butonului//
10 start:
11     if(input(PIN_A4)) //IF=Daca intrarea A4=1 (buton neapasat)...
12     goto start; //mergi la start...
13 //..daca intrarea A4=0 (buton apasat)...
14 //...sari instructiunea dupa if si continua mai jos
15
16     output_high(PIN_B7); //pune 1 logic la iesirea B7
17     goto start; //..si mergi la start.
18
19 } //functia main se termina aici

```

Am utilizat verificarea cu 1 logic fiindca butonul conecteaza intrarea la 0 logic cind este apasat.

Intrarea este la 1 logic prin rezistorul conectat la intrare si +5V cind butonul nu este apasat.

Se putea utiliza si instructiunea **!input** dar trebuia inversata logica in program.

Incercati sa experimentati. Instructiunea **output** va trebui pusa imediat in rindul urmatoare dupa **if**.

Se intelege acum mai bine ce face instructiunea **if**: daca conditia din **if** este adevarata se executa instructiunea imediat urmatoare dupa **if** (sau ce este intre acolade dupa **if**).

Daca conditia din **if** nu este adevarata se sare peste instructiunea imediat urmatoare dupa **if** si se executa in continuare.

Desi nu este necesar se poate folosi si instructiunea **else** =altfel. In acest fel programul poate fi mai clar.

```
1 #include <minicurs_1.h> // #include= include fisier de setari initiale
2
3
4 void main() //functia principala a programului(mereu prezenta)
5 { //incepe dupa aceasta paranteza
6
7
8 // instructiune intrare ////
9 /// aprinde LED B7 la apasarea butonului///
10 start:
11     if(input(PIN_A4)) //IF=Daca intrarea A4=1 (buton neapasat)...
12         goto start; //mergi la start...
13         //..daca intrarea A4=0 (buton apasat)...
14         //...sari instructiunea dupa if si continua mai jos
15     else
16         output_high(PIN_B7); //pune 1 logic la iesirea B7
17     goto start; //..si mergi la start.
18
19 } //functia main se termina aici
```

Dar rulind acest program observam ca LED-ul ramine aprins dupa apasarea butonului si nu se stinge chiar dupa ridicarea sa. Sa il modificam sa se stinga cind butonul nu mai este apasat.

```
1 #include <minicurs_1.h> // #include= include fisier de setari initiale
2
3
4 void main() //functia principala a programului(mereu prezenta)
5 { //incepe dupa aceasta paranteza
6
7
8 // instructiune intrare ////
9 /// aprinde LED B7 la apasarea butonului///
10 start:
11     if(!input(PIN_A4)) //IF=Daca intrarea A4=0 (buton apasat)...
12         output_high(PIN_B7); //pune 1 logic la iesirea B7
13     else
14         output_low(PIN_B7); //pune 0 logic la iesirea B7...
15         //..daca intrarea A4=1 (buton neapasat)...
16
17     goto start; //..si mergi la start.
18
19 } //functia main se termina aici
```

Iar acum sa modificam programul pentru ca LED-ul sa stea aprins o secunda la apasarea butonului.

```
1 #include <minicurs_1.h> // #include= include fisier de setari initiale
2
3
4 void main() //functia principala a programului(mereu prezenta)
5 { //incepe dupa aceasta paranteza
6
7 ///// TTILU SUBRUTINA : aprindere LED 1sec /////
8 // instructiune intrare ///
9 ///// aprinde LED B7 la apasarea butonului///
10 start:
11     if(!input(PIN_A4)) //IF=Daca intrarea A4=0 (buton apasat)...
12     { //remarcati acolada-
13         //executa ce se gasete imediat sub if
14         output_high(PIN_B7); //pune 1 logic la iesirea B7
15         delay_ms(1000); //intrizare 1000mS=1S, LED sta aprins 1 sec
16         output_low(PIN_B7); //pune 0 logic la iesirea B7, LED se stinge
17     } //aici se termina rutina (cu o acolada)
18     //daca A4=1 (buton neapasat)..
19     //nu executa ce e imediat dupa if ci ...
20     goto start; //..executa aici si sare la start.
21
22 } //functia main se termina aici
```

Apasati F1 si cautati instructiunea **input**. Veti gasi mai multe posibilitati de utilizare a acesteia.

4.2 Instructiunea WHILE

Instructiunea **WHILE** se poate folosi ca in exemplul de mai jos (in loc de **IF**):

```
1 #include <minicurs_1.h> // #include= include fisier de setari initiale
2
3
4 void main() //functia principala a programului(mereu prezenta)
5 { //incepe dupa aceasta paranteza
6
7 ///// TTILU SUBRUTINA : aprindere LED 1sec /////
8 // instructiune intrare ///
9 ///// aprinde LED B7 la apasarea butonului///
10 start:
11     while(!input(PIN_A4)) //WHILE= asteapta ca A4 sa fie 0 (buton apasat)...
12     //...si cind e 0 executa in continuare
13     //daca butonul e neapasat A4-1 si sare peste
14     //tot ce este in acolada
15     { //remarcati acolada deschisa
16         output_high(PIN_B7); //pune 1 logic la iesirea B7
17         delay_ms(1000); //intrizare 1000mS=1S, LED sta aprins 1 sec
18         output_low(PIN_B7); //pune 0 logic la iesirea B7, LED se stinge
19         output_toggle(PIN_B6);
20     } //aici se termina rutina (cu o acolada inchisa)
21     goto start; //sare la start.
22
23 } //functia main se termina aici
```

While se mai poate folosi si in alt fel de secvente de programare.

Modificari:

- activati alta iesire decit LED B7 (dupa apasarea butonului ENTER)
- activati concomitent doua iesiri (dupa apasarea butonului ENTER)
- porniti programul activind automat o iesire (de ex LED B7); la apasarea unui buton dezactivati iesirea si activati alta iesire (de ex KB4)
- la apasarea butonului ENTER activati pe rind iesirile cu releu; la apasarea butonului ESC dezactivati pe rind iesirile activate

- dupa ce studiat lectia urmatoare : afisati pe LCD indicativul intrarii activate