

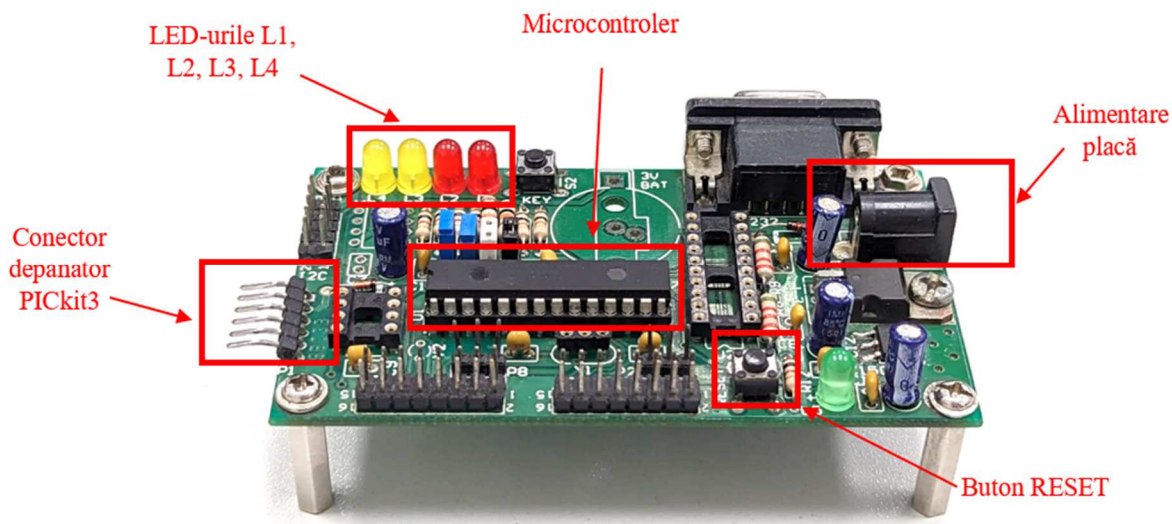
Prezentare placa de dezvoltare si instrumente software

Obiectiv

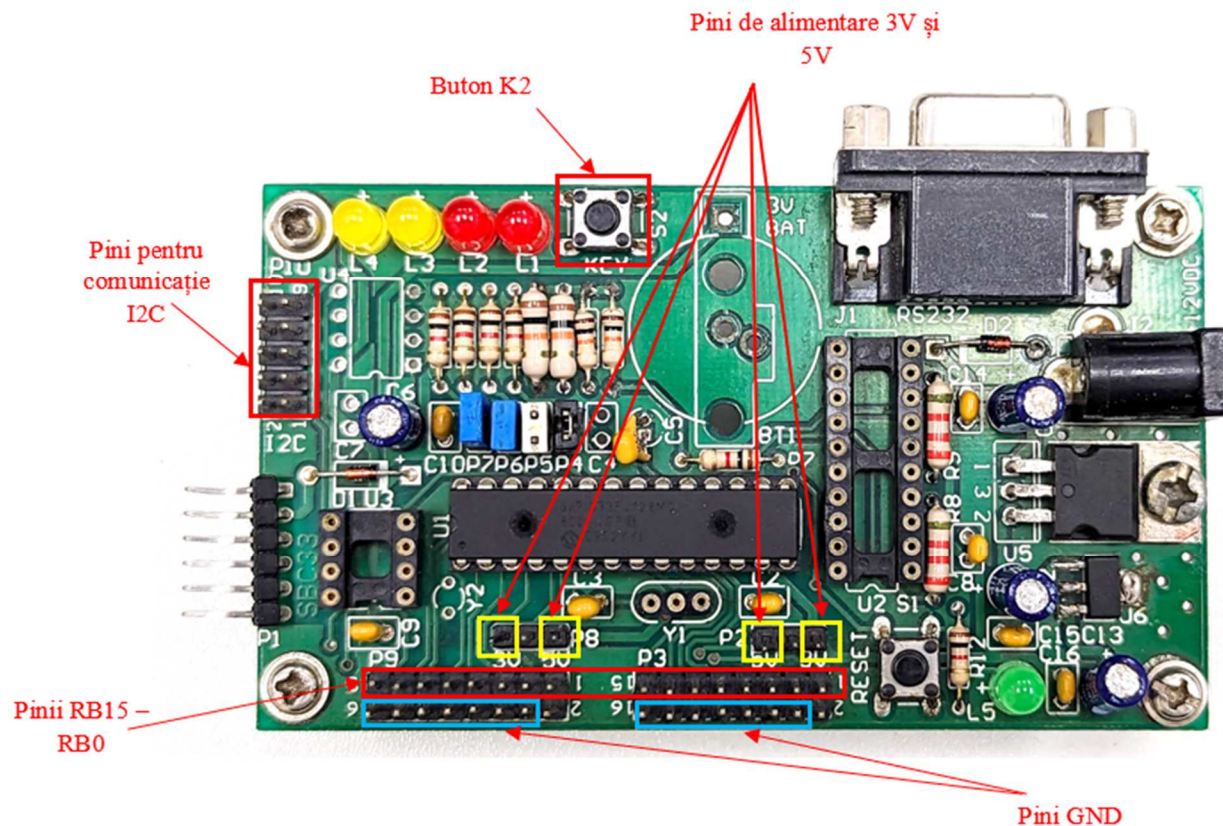
Familiarizarea cu placa de dezvoltare ce va fi folosită pe parcursul ședințelor de laborator și cu instrumentele software folosite (mediul de dezvoltare MPLAB și compilatorul C30).

a) Considerații practice privind placa de dezvoltare cu dsPIC33

În următoarele 2 figuri este prezentată placa de dezvoltare cu microcontrolerul dsPIC.

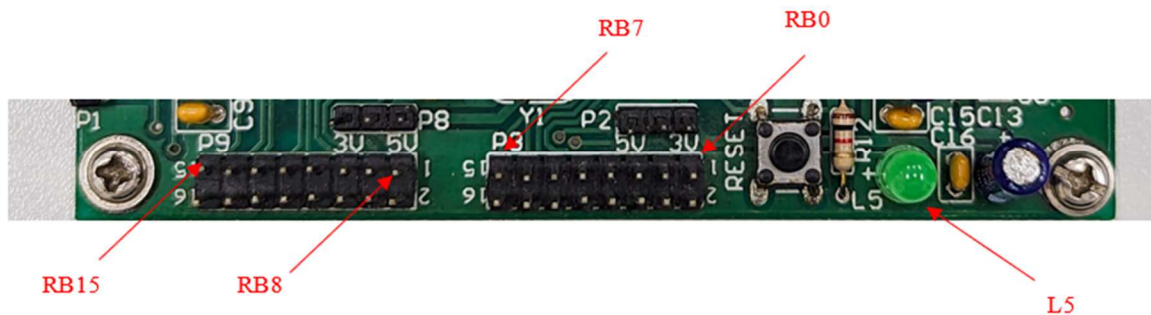


În centrul plăcii de dezvoltare se poziționează **microcontrolerul dsPIC**. Modelele de microcontroller ce pot fi găsite pe plăcile de laborator sunt **dsPIC33FJ64MC802** sau **dsPIC33FJ128MC802**. În partea stânga-sus se află cele **patru LED-uri**, care sunt legate la pinii RB12 – RB15 ai microcontroller-ului. Ele luminează atunci când primesc 0 logic pe pinul microcontroller-ului la care sunt conectate. **Butonul de RESET** este conectat la pinul MCLR (de reset) al microcontrolerului iar acționarea sa va reseta microcontrolerul. **Alimentarea plăcii** de dezvoltare se realizează la o tensiune din domeniul 7.5V-12V, curent continuu. Dacă placa este alimentată, LED-ul verde (L5) se va aprinde.



Butonul K2 este legat la pinul RB7/INT0 al microcontrolerului și poate fi programat să genereze o întrerupere externă. Deși microcontrolerul dsPIC33FJ32MC302 este alimentat la o tensiune de 3,3 V, placa de dezvoltare dispune de 2 perechi de **pini de alimentare**, câte 3V și 5V.

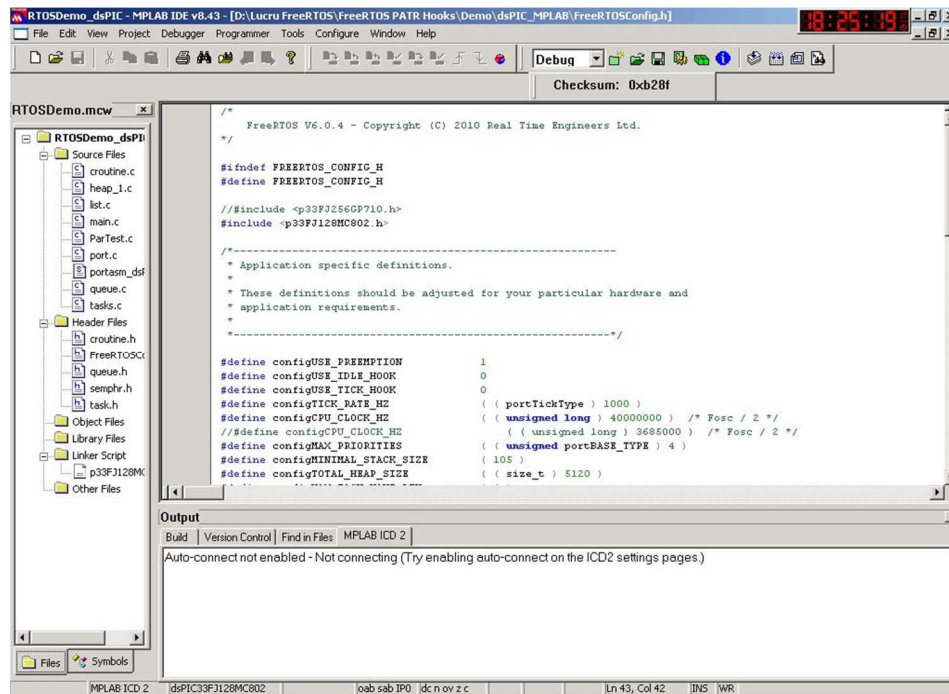
În figura următoare este ilustrat modul de identificare a pinilor ce sunt conectați la pinii PORTB ai microcontroller-ului.



b) Dezvoltarea, compilarea și rularea aplicațiilor în MPLAB

La laborator vor fi utilizate următoarele instrumente software:

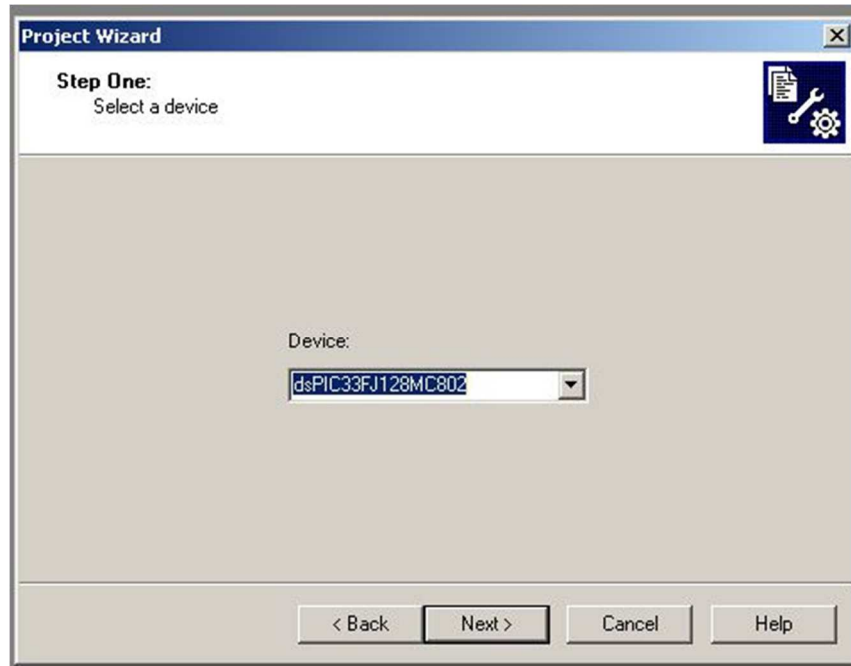
- mediul de dezvoltare MPLAB prezentat în figura de mai jos;
- compilatorul C30 ce reprezintă compilatorul dedicat pentru familia de microcontrollere dsPIC33.



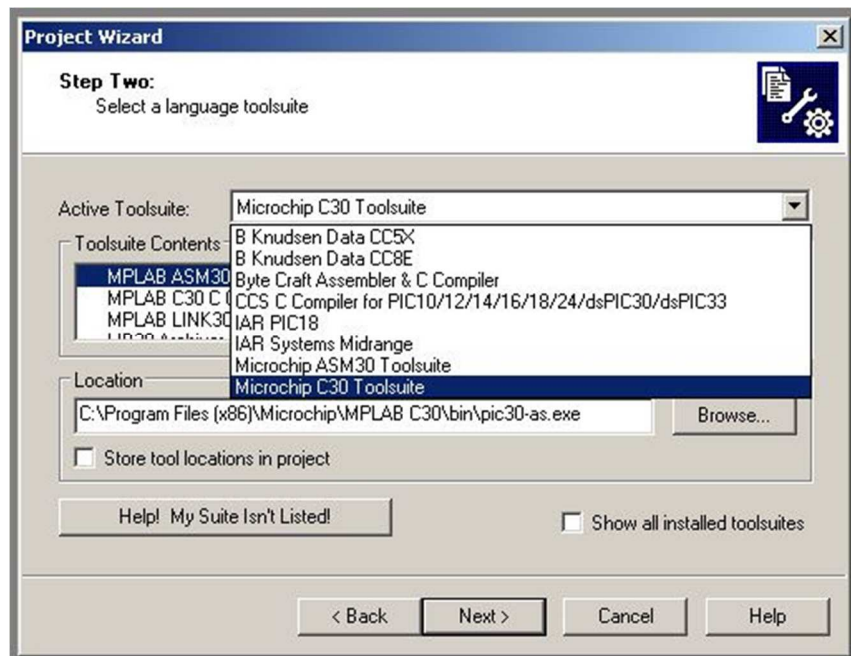
Crearea unui proiect în MPLAB

Crearea unui proiect în MPLAB constă în următoarele etape:

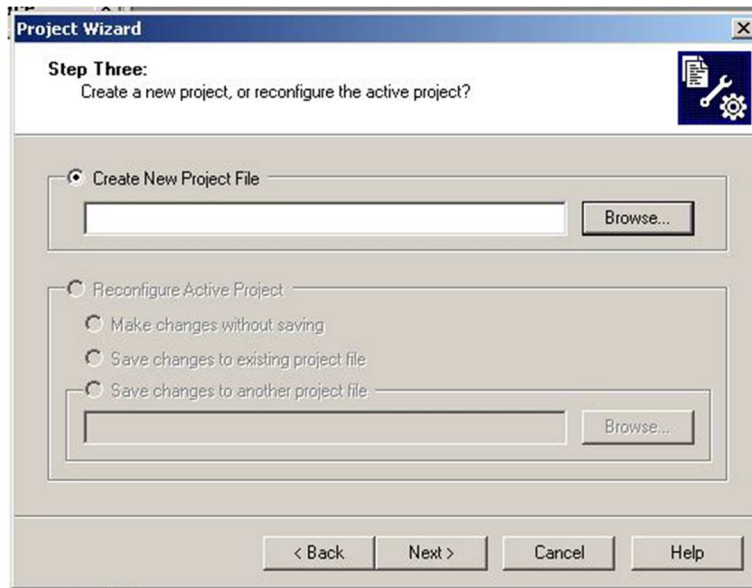
1. se creează un director corespunzător proiectului; denumirea directorului va fi aleasă de către utilizator;
2. se copie în directorul creat fișierul sau fișierele .c și .h ce vor fi incluse în proiect;
3. se creează în MPLAB un proiect folosindu-se opțiunea **Project Wizard** din meniul **Project**).
 - a. Se indică modelul microcontrolerului de pe placa de dezvoltare (dsPIC33FJ64MC802 sau dsPIC33FJ128MC802)



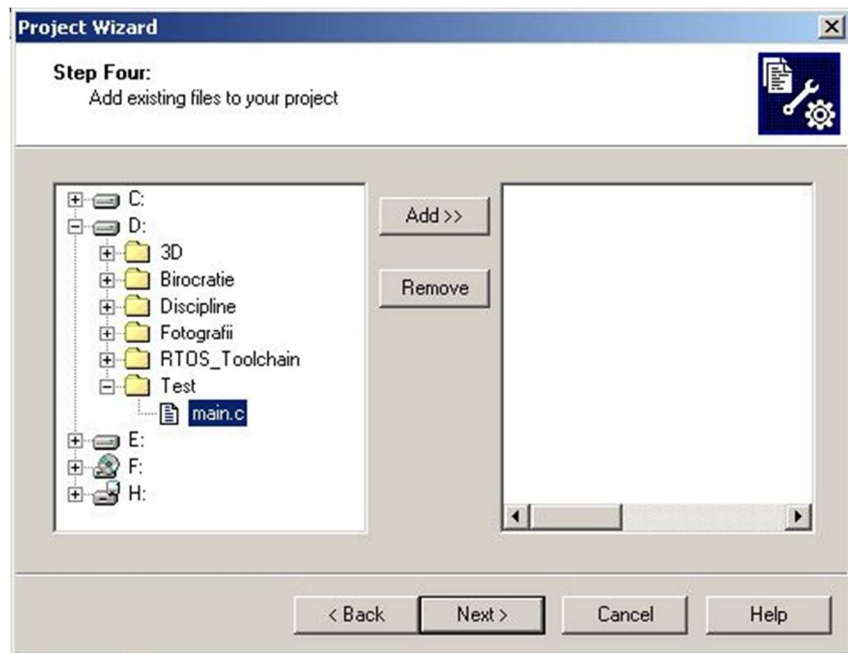
- b. Se alege setul de programe ce va fi folosit (compilerul C30)



- c. Se indică directorul în care va fi salvat proiectul și numele ce va fi dat proiectului (prin acționarea butonului "Browse")



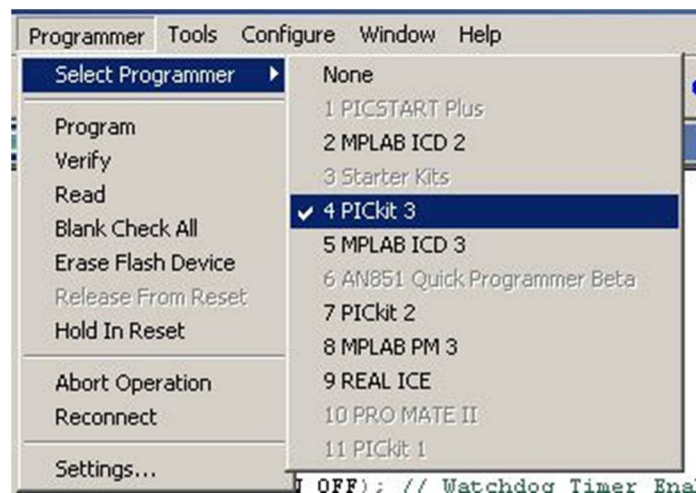
- d. Se adaugă în proiect fișierele deja copiate în directorul acestuia prin selectarea lor și acționarea butonului “Add”;



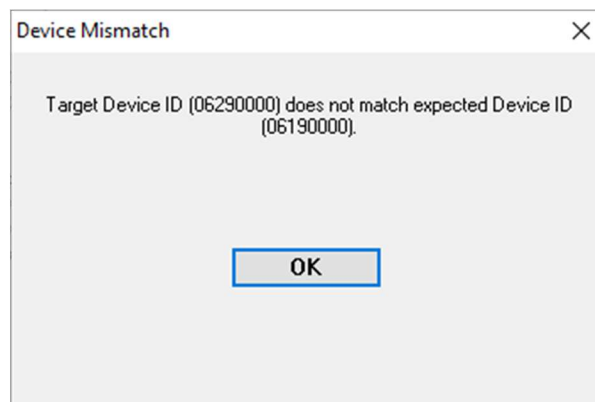
Utilizarea PICkit3 ca programator

Dispozitivul PICkit3 pot fi folosite în două moduri: ca programator (“programmer”) și ca depanator („debugger”). Cu ajutorul său va fi încărcat pe microcontroller programul obținut în urma compilării.

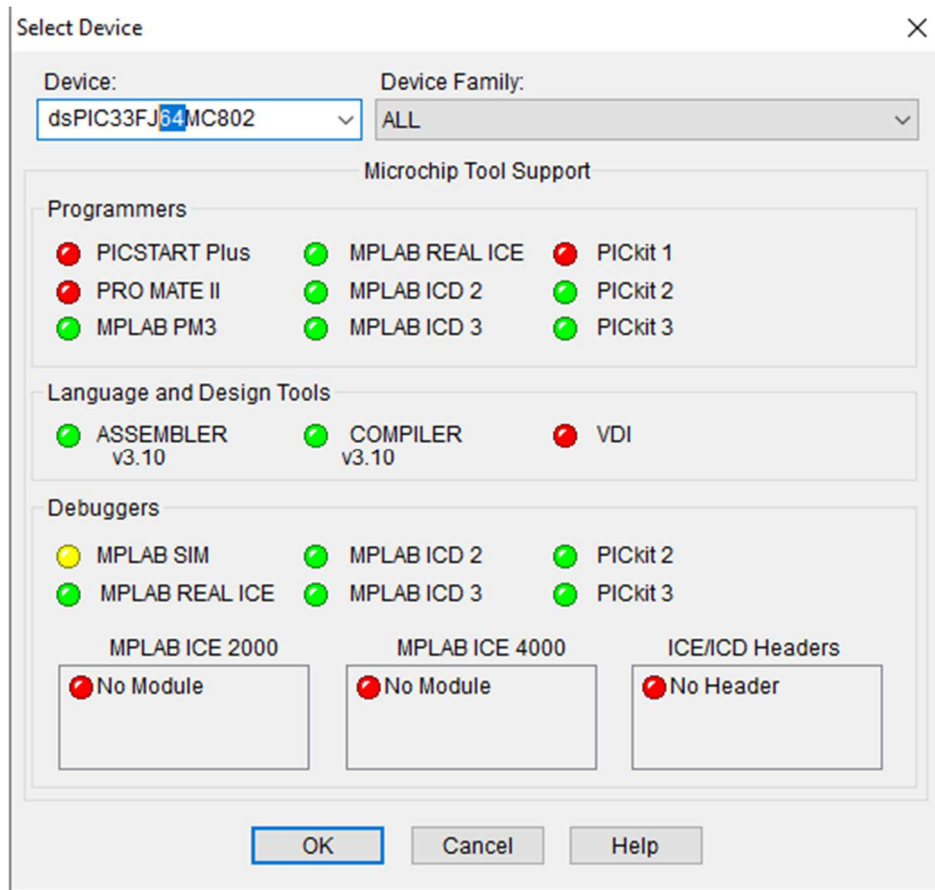
- PICkit3 va fi folosit ca **programator** dacă se selectează opțiunea **Programmer - Select programmer – PICkit 3**, ca în figura de mai jos.



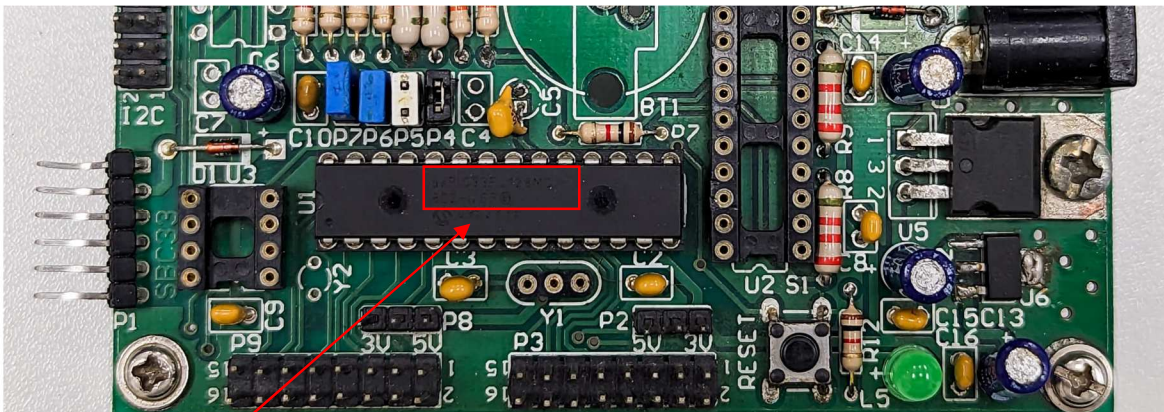
Utilizatorul se poate confrunta cu un mesaj de eroare asemănător celui din imaginea de mai jos la conectarea PICkit3-ului la placa de dezvoltare,:



Acest mesaj indică faptul că modelul de microcontroler indicat în proiect este diferit de modelul de microcontroller existent pe placă. Pentru a rezolva această problemă se va selecta din bara de meniu **Configure – Select device** și se va alege modelul de microcontroler corespunzător plăcii de dezvoltare utilizate, ca în figura următoare.



Modelul microcontroler-ului este inscripționat pe microcontroller.

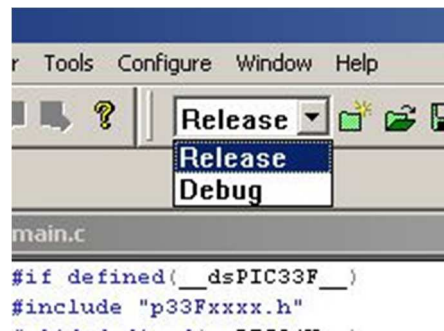


Model microcontroler

După selectarea corectă a dispozitivului, în consolă va apărea următorul mesaj:

```
Output
Build Version Control Find in Files PICKit 3
PICKit 3 detected
Connecting to PICKit 3...
Firmware Suite Version..... 01.25.14
Firmware type.....dsPIC33F/24F/24H
PICKit 3 Connected.
Target Detected
Device ID Revision = 00003004
```

Odată cu selectarea sa ca programator trebuie selectat și modul de compilare corespunzător, și anume modul “**release**”.

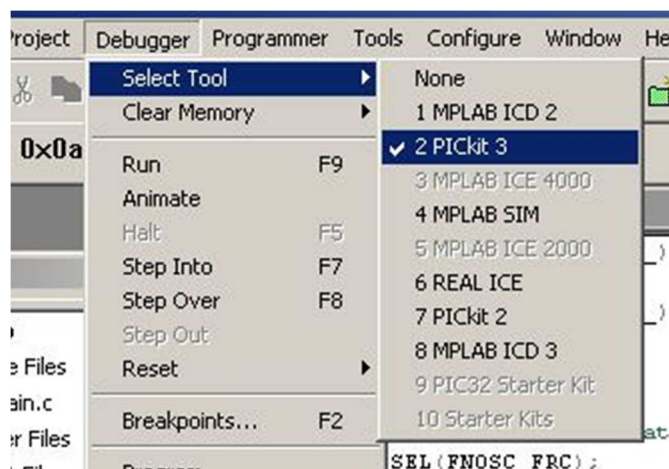


Ca programator, permite compilarea programului (1), încărcarea programului pe microcontroler (2) și pornirea (3), respectiv oprirea (4) programului.

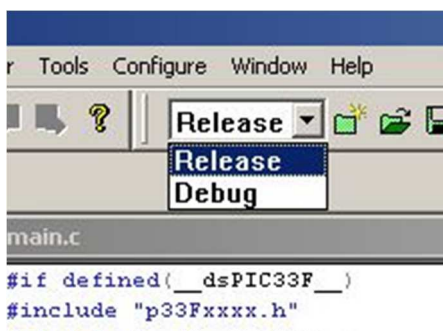


Utilizarea PICKit3 ca depanator

- PICKit3 va fi folosit ca **depanator** dacă se selectează opțiunea **Debugger - Select tool - PICKit 3**.



Odată cu selectarea sa ca depanator trebuie selectat și modul de compilare corespunzător, și anume modul “**debug**”.



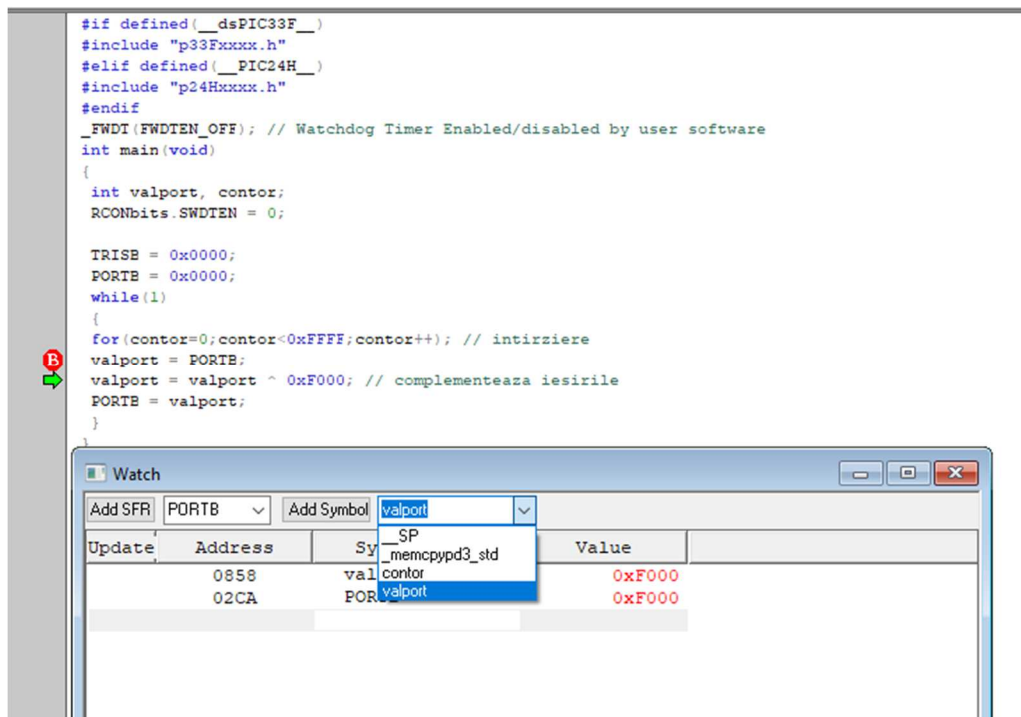
Ca depanator, permite compilarea programului (1), încărcarea programului pe microcontroler (2), resetarea microcontroler-ului (3), pornirea (4) / oprirea (5) programului, dar mai ales depanarea programului prin adăugare de “breakpoints”, rulare pas cu pas (6) și vizualizare de valori pentru regiștri și variabile din program.



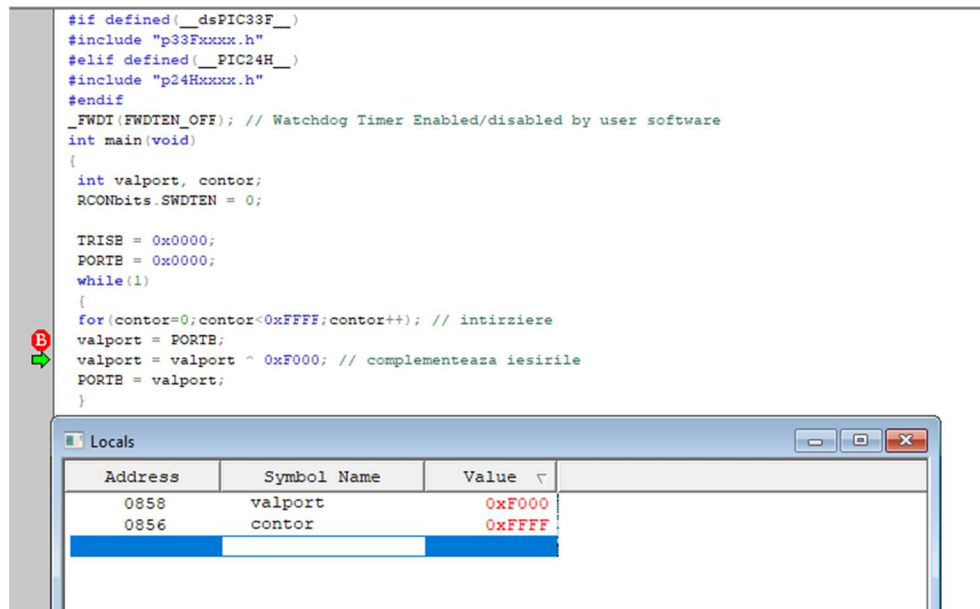
Indiferent de modul în care este setat să lucreze PICKit3, toate operațiunile prezentate până acum pot fi accesate și startate și din meniurile de comandă *Project*, *Programmer* sau *Debugger*.

Vizualizarea valorilor regiștrilor și variabilelor pe parcursul depanării

Valorile regiștrilor de control ai microcontroller-ului și valorile variabilelor globale pot fi vizualizate cu ajutorul ferestrei **Watch**, din meniul **View**. În fereastra **Watch** se pot vedea valorile variabilelor și regiștrilor în locul în care s-a oprit execuția programului sau unde a fost pus breakpoint-ul.



Pentru vizualizarea variabilelor locale se va accesa din meniul **View** fereastra **Locals**. Această fereastră se populează automat cu valorile variabilelor locale ce sunt vizibile în locul în care a fost oprit programul.



Program exemplu

```

#include "p33Fxxxx.h"
int main(void)
{
    int valport, contor;

    TRISB = 0x0000;
    PORTB = 0x0000;

    while(1)
    {
        for(contor=0;contor<0xFFFF;contor++);    // intirziere

        valport = PORTB;
        valport = valport ^ 0xF000;    // complementeaza iesirile
        PORTB = valport;
    }
}

```

În programul de mai sus, este importantă linia **valport = valport ^ 0xF000**, care realizează aprinderea/stingerea celor 4 LED-uri cu ajutorul unui XOR(^). Prin modificarea valorii în hexa se pot obține diferite combinații de aprindere/stingere a LED-urilor.

Exercitii

1. Să se testeze lucrul cu PICkit 3 ca programator și apoi ca depanator
2. Cu PICkit 3 setat ca depanator, să se testeze lucrul cu “breakpoint” și să se vizualizeze valori de variabile și registre.
3. Să se modifice programul de mai sus astfel încât să fie aprinse/stinse doar LED-urile conectate la pinii RB15 și RB13.