

1. Prezentare placă de dezvoltare și instrumente software

a) Scopul lucrării

În cadrul acestei lucrări de laborator se urmăresc următoarele obiective:

- Prezentarea plăcii de dezvoltare și a familiei de microcontrolere dsPIC;
- Prezentarea mediului de dezvoltare MPLAB;
- Programarea și depanarea utilizând microcontrolerul dsPIC33FJ32MC302

b) Considerații practice privind placa de dezvoltare și familia de microcontrolere dsPIC33F

În Figura 1 și Figura 3 este prezentată placa de dezvoltare cu microcontrolerul dsPIC33FJ32MC302.

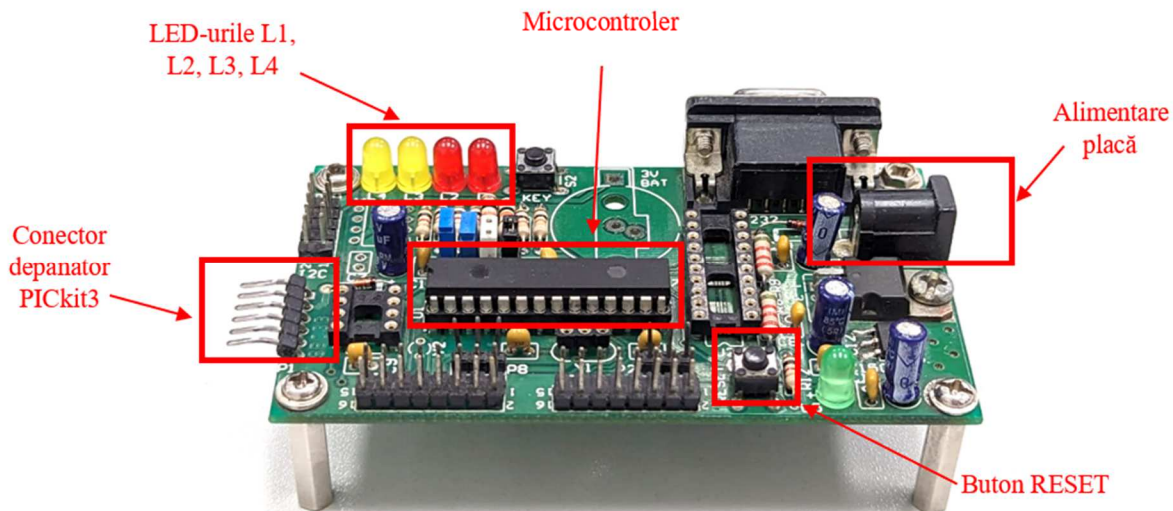


Figura 1. Prezentare LED-uri, butoane, conectori placă de dezvoltare

În centrul plăcii de dezvoltare se poziționează **microcontrolerul dsPIC33FJ32MC302**. În partea de dreapta-sus se află cele **patru LED-uri**, care sunt legate la pinii RB 12 – RB 15 și sunt active pe 0 logic. Acest lucru se poate observa cu ușurință pe schema electrică a plăcii de dezvoltare – LED-urile, care sunt în sine diode, fiind conectate cu anodul la VCC(+), prin rezistențele R1 – R4, când primesc 0V(-) de la pinii microcontrolerului, se creează o diferență de potențial care aprinde LED-ul.

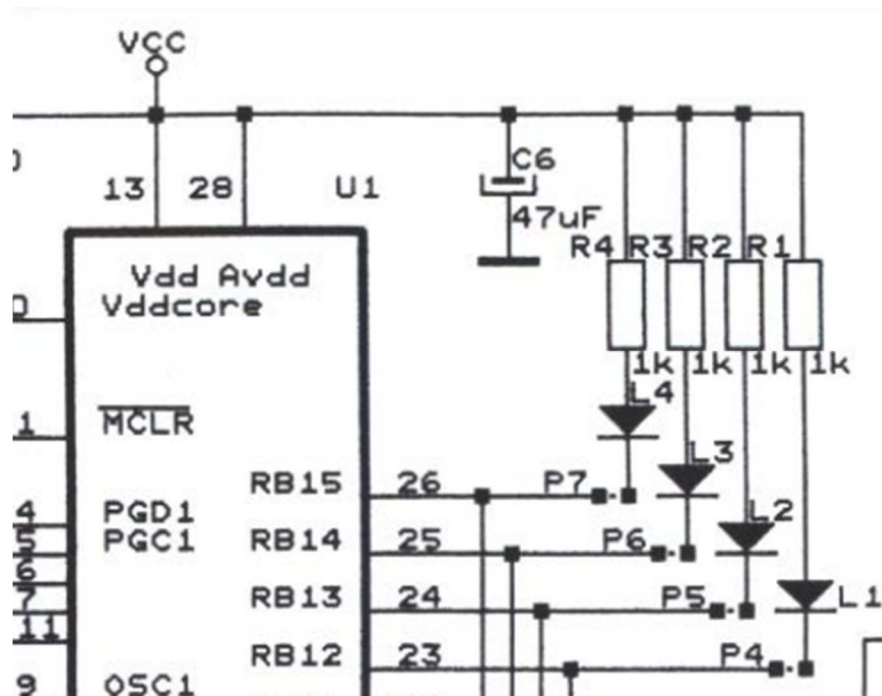


Figura 2. Schema electrică

Butonul de RESET este conectat la pinul MCLR al microcontrolerului, pin active-low ce va reseta microcontrolerul.

Alimentarea plăcii de dezvoltare se realizează la o tensiune de 12V DC. La alimentarea plăcii, LED-ul verde (L5) se va aprinde.

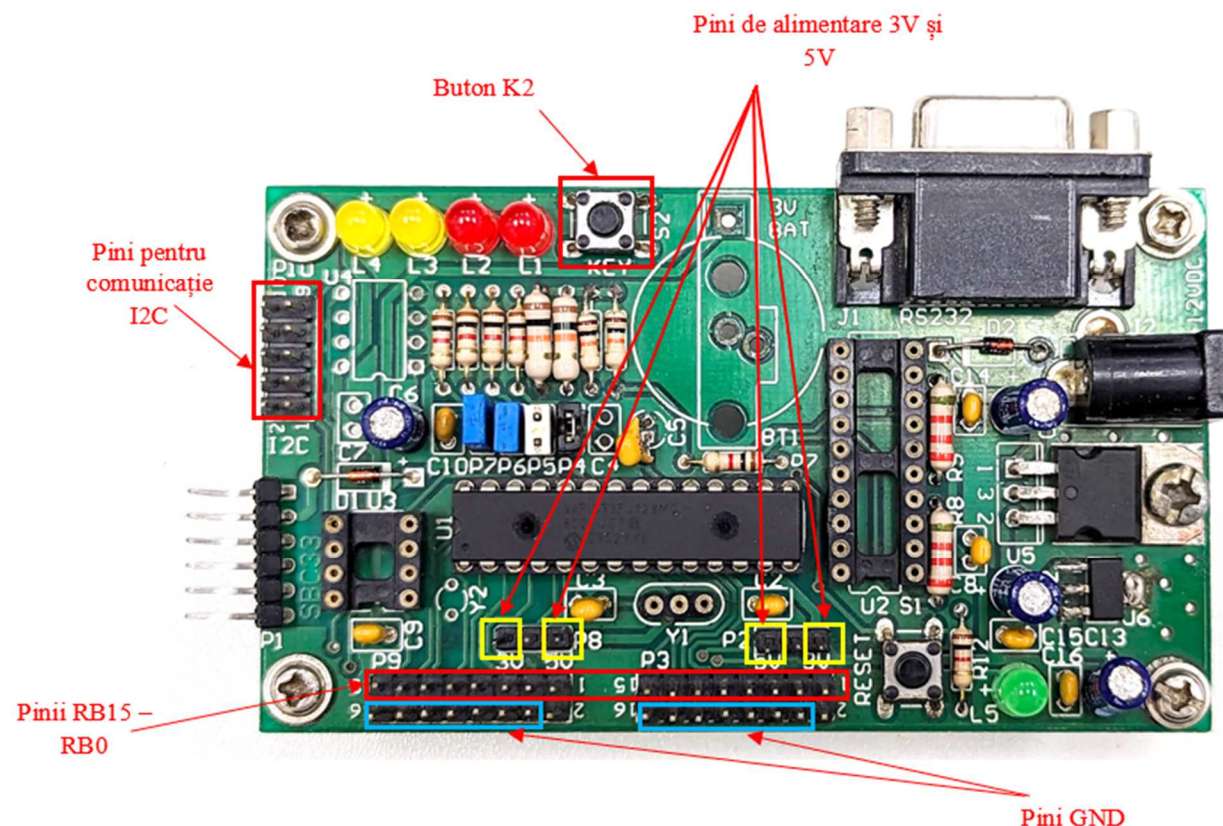


Figura 3. Prezentare pini placa de dezvoltare

Butonul K2 este legat la pinul RB7 al plăcii și la pinul INT0 al microcontrolerului, care reprezintă 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. De asemenea microcontrolerul suportă **modulele de comunicație** CAN, I2C, UART și SPI.

Pinii RB0 – RB15 sunt pini digitali, pe lângă asta mai pot fi folosiți și în alte moduri. Pentru o mai bună înțelegere a modului de utilizare a pinilor RB0-RB15, în tabelul de mai jos vor fi prezentate funcționalitățile acestora.

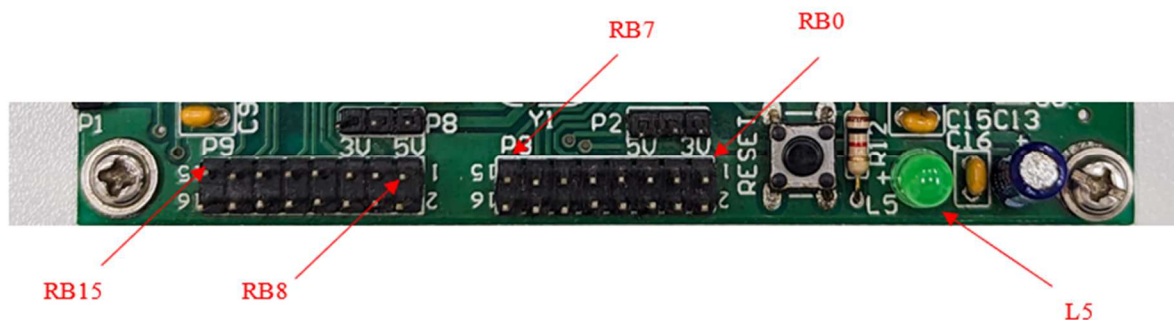


Figura 4. Numerotarea pinilor RB

Pin placă	Pin microcontroler	Descriere
RB0	RP0 / AN2 / PGED1	Pin reconfigurabil / Intrare analogică / Pin I/O de DATE pentru programarea/depanarea canalului de comunicare 1
RB1	RP1 / AN3 / PGEC1	Pin reconfigurabil / Intrare analogică / Pin de intrare al CEASului pentru programarea/depanarea canalului de comunicare 1.
RB2	RP2 / AN4	Pin reconfigurabil / Intrare analogică
RB3	RP3 / AN5	Pin reconfigurabil / Intrare analogică
RB4	RP4	Pin reconfigurabil
RB5	RP5 / PGED3	Pin reconfigurabil / Pin I/O de DATE pentru programarea/depanarea canalului de comunicare 3
RB6	RP6 / PGEC3 / ASCL1	Pin reconfigurabil / Pin de intrare al CEASului pentru programarea/depanarea canalului de comunicare 3 / I/O alternativ a ceasului serial sincron pentru I2C
RB7	RP7 / INT0	Pin reconfigurabil / Întrerupere externă
RB8	RP8 / PWM / SCL1	Pin reconfigurabil / Pin PWM / I/O a ceasului serial sincron pentru I2C
RB9	RP9 / PWM / SDA1	Pin reconfigurabil / Pin PWM / I/O de date seriale sincrone pentru I2C

Laborator 1Prezentare placă de dezvoltare

RB10	RP10 / PGED2 / PWM	Pin reconfigurabil / Pin I/O de DATE pentru programarea/depanarea canalului de comunicare 2 / Pin PWM
RB11	RP11 / PGEC2 / PWM	Pin reconfigurabil / Pin de intrare al CEASului pentru programarea/depanarea canalului de comunicare 2 / Pin PWM
RB12	RP12 / PWM	Pin reconfigurabil / Pin PWM
RB13	RP13 / PWM	Pin reconfigurabil / Pin PWM
RB14	RP14 / PWM / RTCC	Pin reconfigurabil / Pin PWM / Ieșire alarmă RTC
RB15	RP15 /PWM	Pin reconfigurabil / Pin PWM

Tabel 1. Descriere pini placă de dezvoltare

2. 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;
- particularizarea de compilator C pentru familia de microcontrollere dsPIC33 și anume C30, ce este integrat în MPLAB.

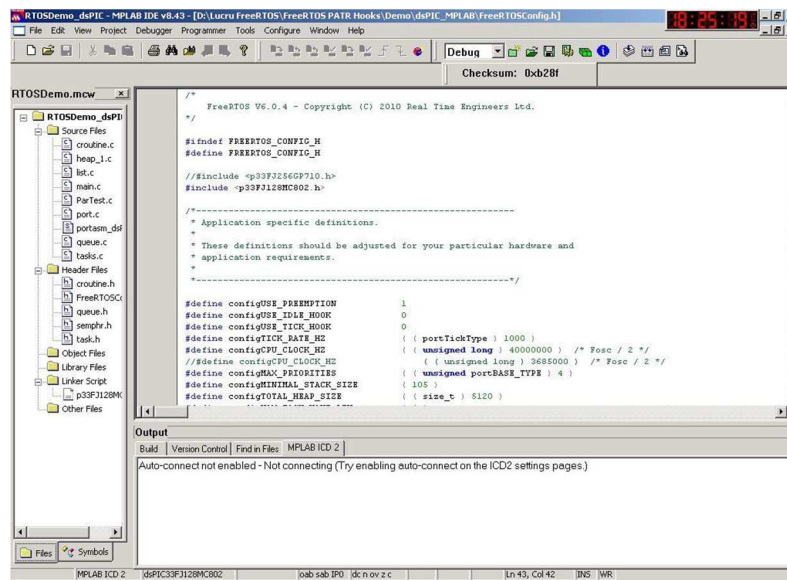


Figura 5. Mediul de dezvoltare MPLAB

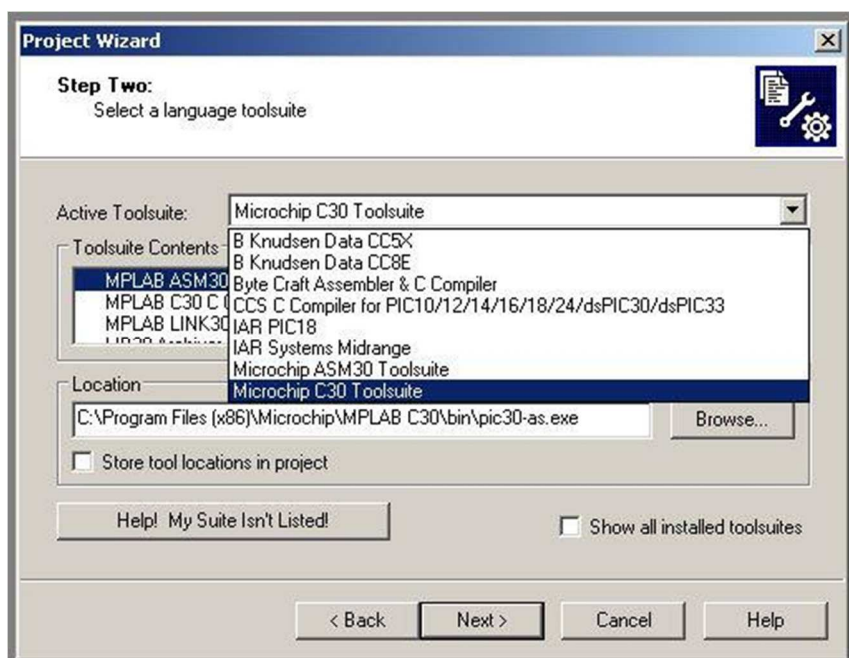
Crearea unui proiect în MPLAB constă în următoarele etape:

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)

Laborator 1 Prezentare placă de dezvoltare

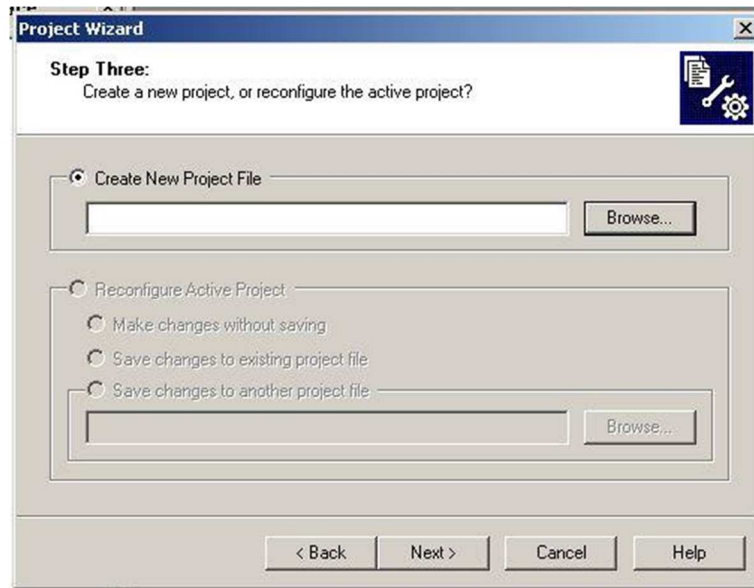


- b. Se alege setul de programe ce va fi folosit (compilerul C30)

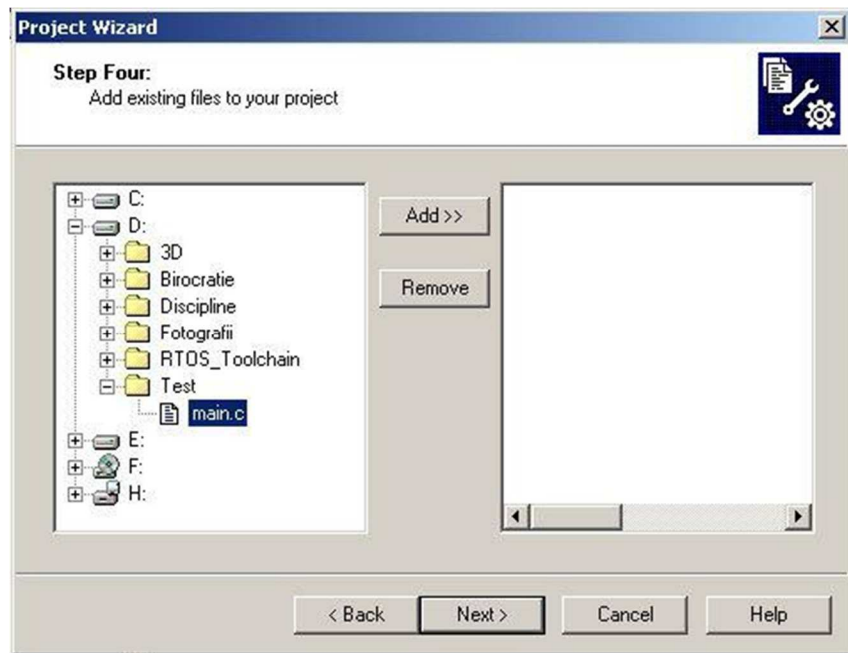


- c. Se indică directorul în care va fi salvat proiectul și numele ce va fi dat proiectului (prin acționarea butonului "Browse")

Laborator 1 Prezentare placă de dezvoltare



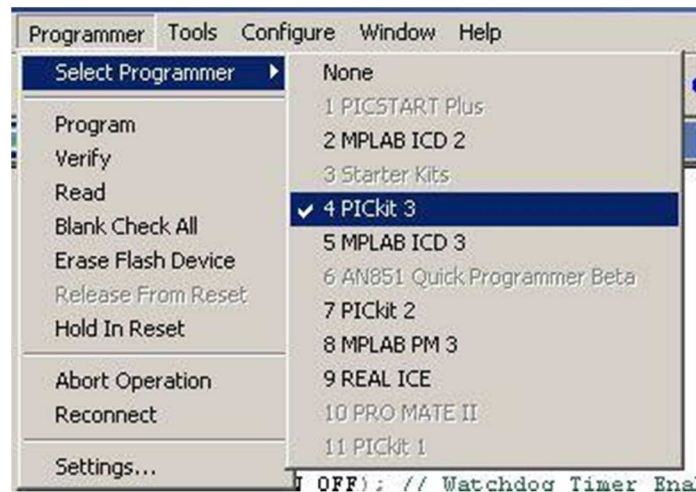
- d. Se adaugă în proiect fișierele deja copiate în directorul acestuia prin selectarea lor și acționarea butonului “Add”;



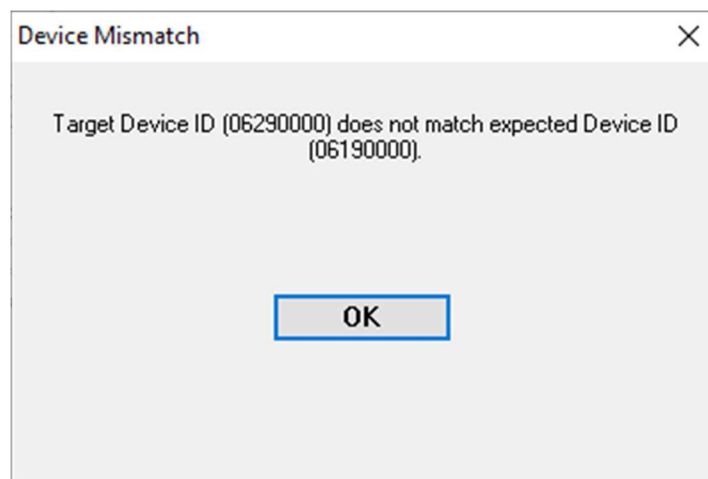
Dispozitivul PICkit3 pot fi folosite în două moduri: ca programator (“programmer”) și ca depanator („debugger”).

Laborator 1 Prezentare placă de dezvoltare

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

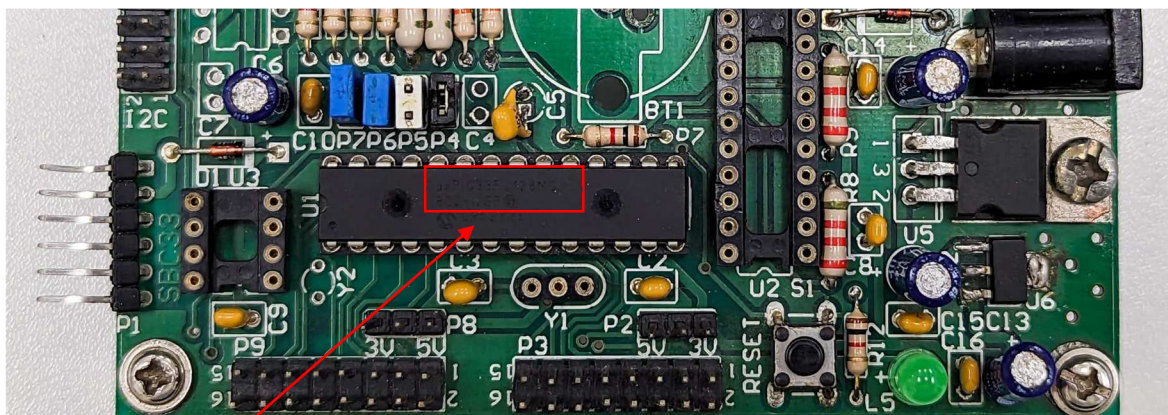


La încercarea de a se conecta la dispozitivul PICkit3, utilizatorul se poate confrunta cu un mesaj de eroare asemănător celui din imaginea de mai jos:

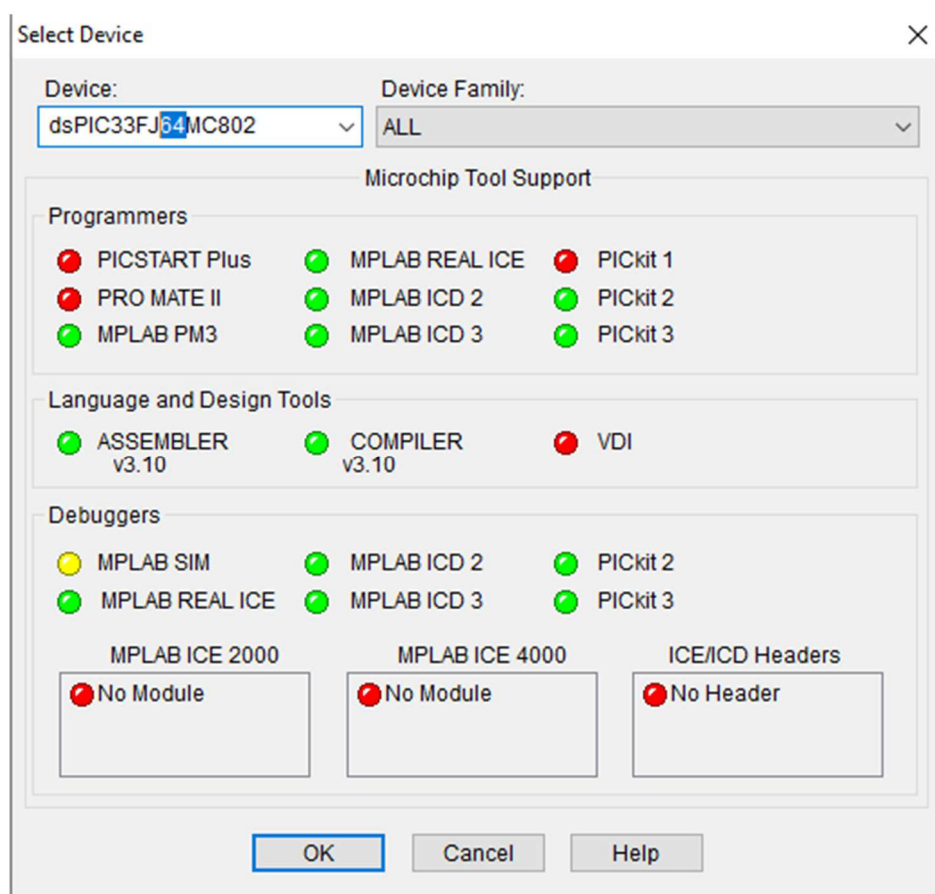


Acesta semnifică faptul că a fost selectat un model diferit de microcontroler în timpul configurării. Pentru a rezolva problema respectivă, se va selecta din bara de meniu **Configure – Select device** și se va alege modelul de microcontroler corespunzător plăcii de dezvoltare utilizate. Pentru a afla modelul de microcontroler, e suficient să îl citim de pe placa de dezvoltare.

Laborator 1Prezentare placă de dezvoltare

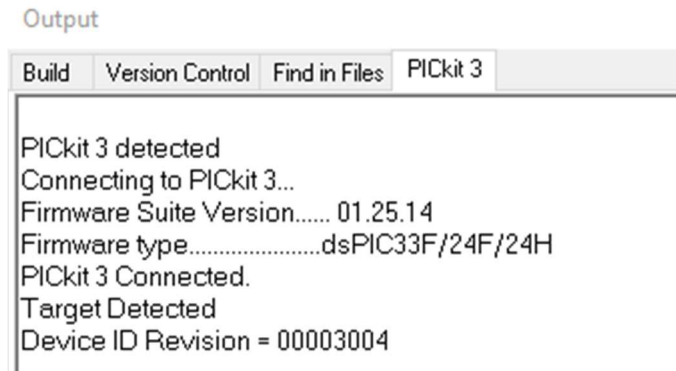


Model microcontroller

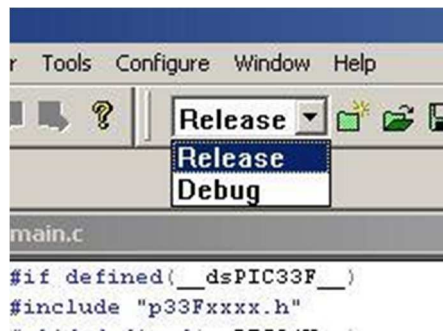


După selectarea corectă a dispozitivului, în consolă va apărea următorul mesaj:

Laborator 1 Prezentare placă de dezvoltare



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.

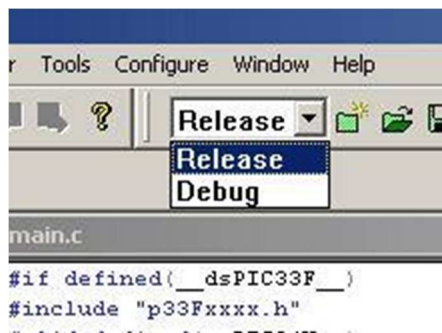


Laborator 1 Prezentare placă de dezvoltare

- 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**”.



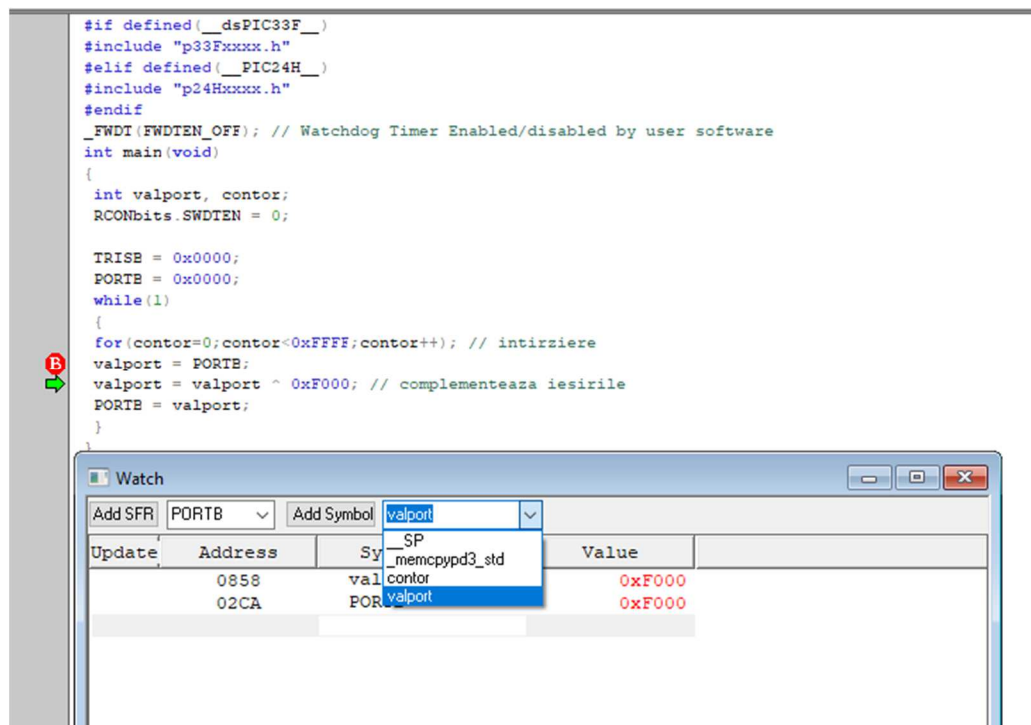
Ca depanator, permite compilarea programului (1), încărcarea programului pe microcontroler (2), resetarea microcontroler-ului (3), pornirea (4) / oprirea (5) programului, dar mai ales depănarea programului prin adăugare de “breakpoints”, rulare pas cu pas (6) și vizualizare de valori pentru regiștri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



Compilarea și rularea aplicațiilor dezvoltate în limbajul C în mediul de dezvoltare MPLAB pentru microcontrolere din familia dsPIC33 pot fi startate și din meniurile de comandă.

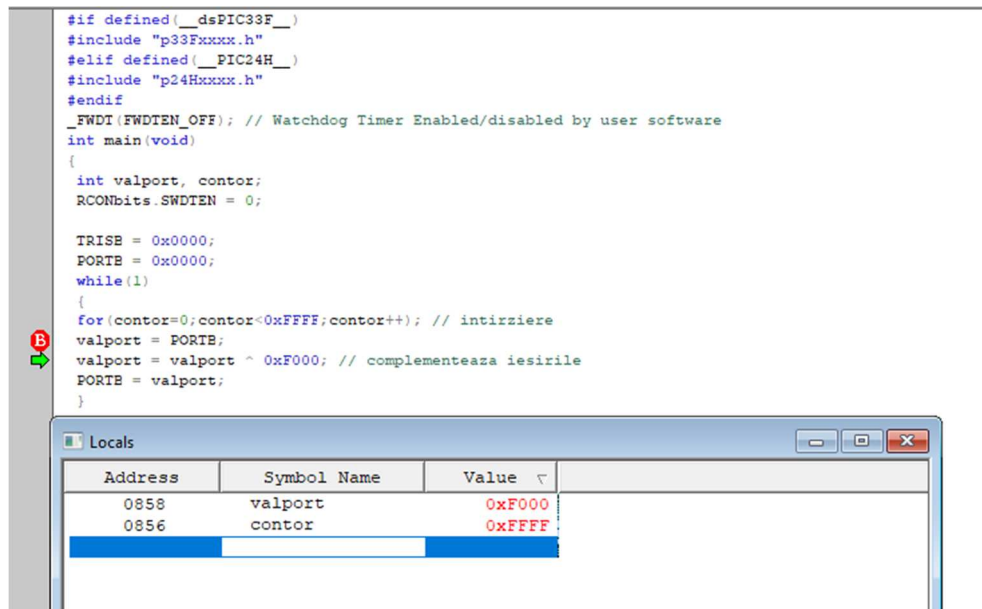
Laborator 1 Prezentare placă de dezvoltare

1. Compilarea - **Project - Build All**
2. Se încarcă programul pe microcontroler: **Debugger – Program** sau **Programmer - Program**
3. În funcție de modul de lucru, care poate fi programare sau depanare, se lansează în execuție aplicația din meniul **Programmer** cu **Release from Reset** sau se urmărește execuția folosind din meniul **Debugger** opțiunile: **Program - Halt, Breakpoint**, etc, eventual cu vizualizare valori variabile **View - Watch**. În fereastra Watch se pot vizualiza valorile variabilelor și porturilor în locul în care s-a pus breakpoint-ul.

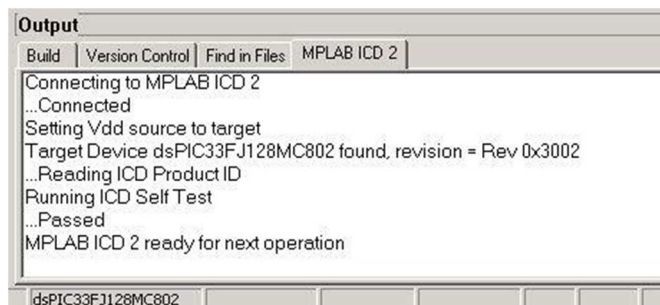


Pentru vizualizarea regiștrilor se va accesa din **View** fereastra **Locals**.

Laborator 1 Prezentare placă de dezvoltare



4. Se poate verifica setarea privind microcontrolerul folosit în meniul **Configure - Select Device**: dsPIC33FJ128MC802 sau dsPIC33FJ64MC802
5. Se verifică conectarea la depanator/programator cu: **Debugger – Connect**, respectiv **Programmer – Connect**. În fereastra Output trebuie să fie afișate informațiile privind conectarea (figura de mai jos).



Program exemplu

```
#if defined(__dsPIC33F__)
#include "p33Fxxxx.h"
#elif defined(__PIC24H__)
#include "p24Hxxxx.h"
#endif

_FWDT(FWDTEN_OFF); // Watchdog Timer Enabled/disabled by user software

int main(void)
{
    int valport, contor;
    RCONbits.SWDTEN = 0;

    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, esențială ar fi linia **valport = valport ^ 0xF000**, care definește activarea și dezactivarea celor 4 LED-uri, cu ajutorul unui XOR(^). Dacă e să facem conversia din hexa în binar: F = 1111, care reprezintă de fapt cele 4 diode. Prin modificarea valorii în hexa putem obține diferite combinații de activare/dezactivare 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” si sa se vizualizaze valori de variabile si registri.
3. Să se modifice programul de mai sus astfel încât să fie activate/dezactivate doar LED-urile conectate la pinii RB15 și RB13.