

Lucrarea nr. 3 Tehnici de programare a aplicațiilor pe sisteme cu microcontroloare

1. Obiectivul lucrării

Lucrarea urmărește studierea tehnicilor de dezvoltare și depanare a programelor scrise pentru sisteme de calcul dedicate; studierea mediului de dezvoltare MPLAB IDE (integrated development Environment) ca mijloc de proiectare, implementare și testare a aplicațiilor pentru microcontroloare din familia PIC.

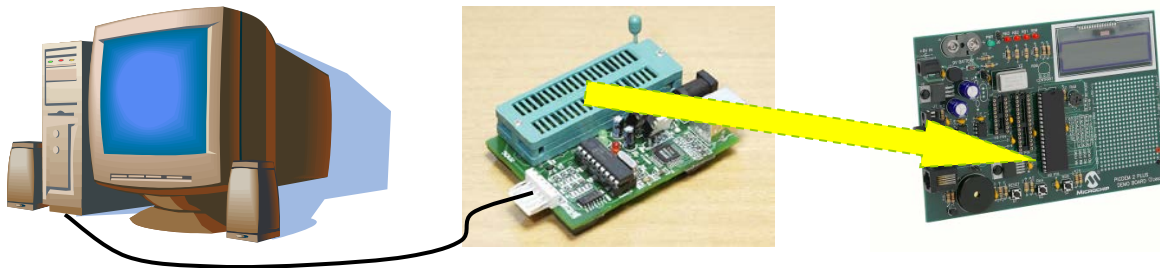
2. Considerații teoretice

2.1 Tehnici de programare și depanare a aplicațiilor dezvoltate pe sisteme cu microcontroloare

Ca metodologie de lucru dezvoltarea aplicațiilor pe platforme încapsulate, bazate pe microcontroloare diferă semnificativ de metodologia de dezvoltare a aplicațiilor pentru calculatoare personale. În general sistemele bazate pe microcontroloare nu dispun de resursele necesare pentru dezvoltarea de aplicații (memorie, tastatură, afișaj, sistem de operare, sistem de gestiune a fișierelor, etc.). De aceea dezvoltarea aplicațiilor pentru aceste sisteme se realizează pe un sistem de calcul uzual (ex: PC), urmând ca programul executabil generat în urma compilării să fie transferat către sistemul țintă (target system) sau înscris direct în memoria microcontrolorului.

Anumite firme (inclusiv cele care furnizează circuitele) oferă diferite instrumente de proiectare, programare și testare a sistemelor încapsulate bazate pe microcontroloare. În funcție de serviciile oferite și de complexitatea acestora putem identifica următoarele categorii de instrumente de dezvoltare:

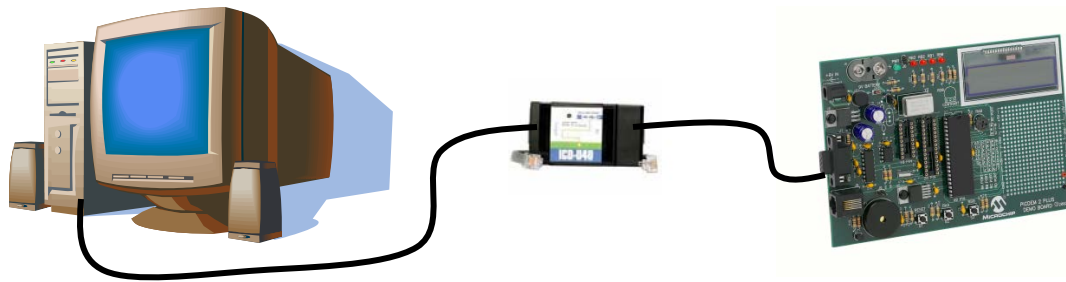
- programator de microcontroloare – ansamblu hardware-software relativ simplu care oferă funcții minime necesare pentru programarea (inscrierea) aplicațiilor în memoria internă a microcontrolorului; nu oferă facilități de dezvoltare a programelor sau de depanare a acestora



- starterkit – ansamblu minimal necesar pentru programarea și testarea programelor; se oferă o schemă generală simplă, utilă în verificarea și testarea unei familii de microcontroloare și instrumente de programare care includ un asamblor și un link-editor



- mediu integrat de dezvoltare IDE – o aplicatie complexa ce permite parcurgerea principalelor etape de proiectare a unei aplicatii, de la editarea programului, compilare, link-editare, transferul catre un sistem tinta si executia programului; mediul poate sa includa un simulator de microcontrolor, caz in care executia aplicatiei se poate face pe calculatorul personal; in cazul microcontroloarelor PIC firma Microchip ofera in acest scop produsul MPLAB IDE
- ICD – in-circuit-debugger – circuit ce permite depanarea si testarea programelor in schema finala a aplicatiei (in sistemul tinta)



- ICE – in-circuit-emulator – un dispozitiv ce permite ca PC-ul sa ia locul microcontrolerului din schema tinta si emularea functionarii acestuia; este cel mai performant instrument de testare a sistemelor incapsulate; programatorul poate sa urmareasca in detaliu functionarea sistemului in diferite regimuri de lucru (pas-cu-pas, automat, etc.) si pot fi controlate componentele microcontrolerului (registre generale si speciale, interfete, etc.).



2.2 Mediul de dezvoltare MPLAB IDE

MPLAB IDE este un mediu integrat de dezvoltare (IDE – Integrated Development Environment) oferit de firma Microchip pentru familiile de microcontroloare PIC12, PIC16, PIC18, etc. Include:

- manager de proiecte
- editor de program sursa
- asamblor
- editor de legături
- depanator (debugger)
- motor de executie

Aplicatia permite integrarea si a altor componente (ex: compilator C, depanator, programator, ICD, ICE) realizate fie de firma Microchip fie de alte firme (ex. CCS).

Figura de mai jos indica schema de administrare a unui proiect prin MPLAB:

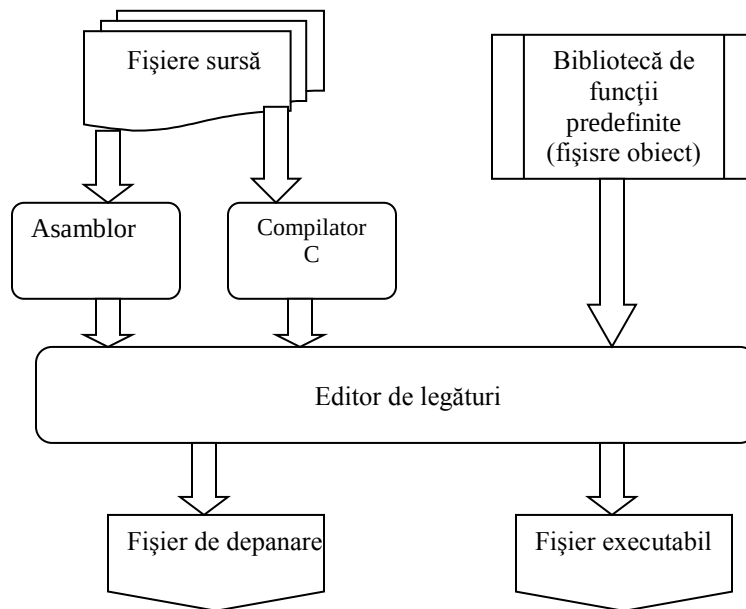


Figura 1. Administrarea unui proiect în MPLAB

Pasii necesari pentru dezvoltarea unei aplicatii:

a. Lansarea mediului MPLAB IDE

- *Start>Programs>Microchip>MPLAB IDE vx.xx>MPLAB IDE*

b. Selectarea dispozitivului

- trebuie să preceadă crearea unui nou proiect
- din meniu se alege: *Configure->Select device*
- se alege tipul circuitului folosit (ex: *PIC16F877*); mediul va indica tipurile de dispozitive suportate pentru programarea circuitului (ex: MPLAB ICD 2), limbajele acceptate și instrumentele de depanare utilizabile

c. Crearea unui proiect - cu ajutorul "project wizard"

- din meniu se alege: *Project>Project Wizard*

- în pași următori se aleg: tipul de dispozitiv și setul de instrumente de compilare și link-editare; se observă că pe lângă instrumentele Microchip pot fi folosite instrumente oferite de alți producători (ex: compilator C al firmei HI-TECH)
- se alege un nume pentru noul proiect (ex: Test_x)
- se adaugă fișiere sursă; pentru început se poate adăuga un fișier “model” (template) care conține elementele inițiale de configurare pentru fiecare circuit în parte (ex: C:\Program Files\Microchip\MPASM Suite\Template\Object\16F877tmpo.asm); se apasă litera de la începutul fișierului până apare “C” care va copia fișierul în proiect
- se adaugă un fișier necesar pentru link-editare (ex: C:\Program Files\Microchip\MPASM Suite\LKR\16F8722.lkr sau \16F8722i.lkr dacă se folosește ICD 2)
- proiectul creat se poate vizualiza cu: *View>Project.*; fișierele pot fi selectate pentru editare (dublu-click pe fișier sau click-dreapta și Edit)
- se construiește proiectul (asamblare, link-editare și conversie în format HEX) cu: *Project>Build All* sau prin apăsarea icoanei corespunzătoare la “Build All”
- pentru scriere de program se deschide fișierul model (dublu-click pe 16F877tmpo.asm) și se adaugă instrucțiuni în “Main” după comentariul: *remaining code goes here* ; se va adăuga următorul cod:

```

    clrf w_temp
    clrf w_temp
    BCF  STATUS, RP0 ; comuta pe bancul 0
    BCF  STATUS, RP1
    movwf PORTA      ; clear PORTA
    BSF  STATUS, RP0 ; comuta pe bancul 1
    MOVLW 0xF0       ;
    movwf TRISA      ; configure PORTA D7-4 input, D3-0=output
    bcf STATUS, RP0

```

Init

```

    clrf COUNT      ; initialize counter

```

IncCount

```

    incf COUNT,F
    movf COUNT,W    ; increase count and
    movwf PORTC     ; display on PORTC
    call Delay      ; go to Delay subroutine
    goto IncCount   ; infinite loop

```

Delay

```

    movlw 0x40
    movwf DVAR2     ; set outer delay loop

```

DelayOuter

```

    movlw 0xFF
    movwf DVAR      ; set inner delay loop

```

DelayInner

```

    decfsz DVAR,F
    goto DelayInner
    decfsz DVAR2,F

```

```
goto DelayOuter  
return
```

- în zona de declarare a variabilelor se va adăuga:

```
COUNT RES 1      ; Counter  
DVAR RES 1        ; inner loop counter  
DVAR2 RES 1       ; outer loop counter
```

- se reconstruiește proiectul și se corectează eventualele erori

d. Testarea codului – cu ajutorul simulatorului sau a unui sistem “tintă”, cum ar fi de exemplu IDC 2 (in-circuit debugger) sau ICE (in-circuit emulator);

- alegerea instrumentului de testare se face prin : *Debugger>Select Tool>MPLAB ICD 2 sau Debugger>Select Tool>MPLAM SIM* (pentru simulator);

- în primul caz conectarea la ICD 2 se face prin: *Debugger>Connect*; în caz de eroare se parcurge *Debugger>MPLAB ICD2 setup wizard*

- rularea programului în regim continuu sau pas-cu-pas se face cu comenzile din Debugger: RUN (F9), Step into(F7), Step Over(F8), Step Out și Reset; pentru rularea cu puncte de întrerupere se folosesc facilitățile “Breakpoint”.

3. Modul de desfășurare a laboratorului

3.1 Se va instala mediul de dezvoltare MPLAB de pe CD-ul furnizat de firma Microchip sau de pe pagina web a firmei (www.microchip.com)

3.2 Se vor urma pașii de dezvoltare indicați în Manualul de utilizare a aplicației MPLAB (anexa: MPLAB.pdf)

3.3 Se va testa funcționarea aplicației create cu ajutorul simulatorului încorporat în mediul de depanare; se vor testa diversele facilități de depanare:

- executia pas-cu-pas si în regim automat (step-in, step-out, Run, stop, Reset, etc.)
- vizualizarea variabilelor (watch)

3.4 Se va testa funcționarea aplicației create cu ajutorul circuitului MPLAM ICD 2;

- se conectează circuitul ICD 2 la calculator
- se conectează placa țintă (ex: PICDEM 4) la ICD 2
- se încarcă programul în microcontrolorul de pe placa țintă
- se execută aplicația în regim pas-cu-pas și continuu