9.Sistemul de dezvoltare

Cuprins

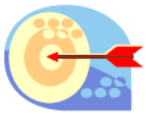
Cuprins Laborator 9

Cunoştinţe preliminare

Limbaj de asamblare MCS 51

Introducere

Prima lucrare de interfaţare cu microcontrollere prezintă sistemul de dezvoltare IMC500 pentru familia MCS 51. Este prezentată placa, microcontrollerul 80C552 cu magistrală externă, schema electrică de conectare, porturile de I/O, resursele interne şi modul de generare a semnalelor de selecţie. La partea software sunt prezentate formatul Intel HEX şi programul monitor a plăcii de dezvoltare. Practic se realizează scrierea unui program simplu, testarea lui şi înscrierea în microcontroller.



Obiective

După parcurgerea acestui modul studenţii vor înţelege:
Resursele interne ale microcontrollerului;
Modul de scriere a unui program şi înscrierea lui în microcontroller;
Verificarea programului şi simularea funcţionării;
Testarea programului înscris cu ajutorul programului monitor.

Durata medie de studiu individual

Durata medie de studiu individual este de 2 ore.

9.1. Sistem de dezvoltare cu microcontroler 80C552

Sistemul **IMC500**, a fost astfel conceput încât permite dezvoltarea rapidă a aplicațiilor în domenii diverse – automatizări industriale, aparate de măsură, industrie uşoară, medicină, industria automobilelor, domeniul casnic, etc. Utilizarea tehnologiei **CMOS** îl recomandă pentru aplicațiile care necesită un consum redus de energie şi care necesită imunitate ridicată la perturbații. În figura 9.1 este prezentată partea plantată a sistemului.

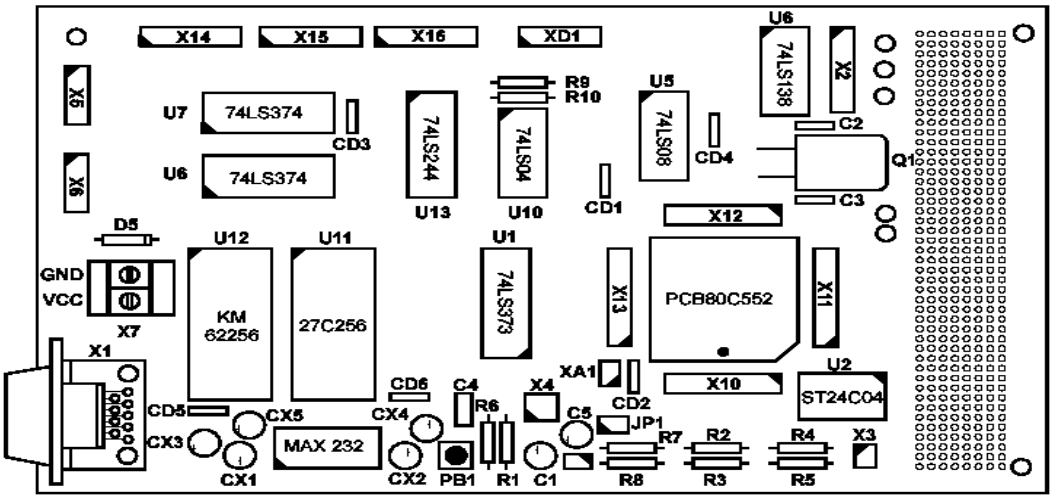


Figura 9.1. Sistemul de dezvoltare IMC500

Acest sistem de dezvoltare, bazat pe microcontroller-ul 80C552, acoperă aplicațiile dezvoltate cu microcontrollerele **80C31**, **80C32** şi alte microcontrollere din familia **8051**, putând fi folosit la dezvoltarea de aplicații. Sistemul de dezvoltare dispune de următoarele resurse hardware şi caracteristici tehnice (figura 9.2):

- microcontroler PCB80C552 (fără memorie internă de program), lucrând la o frecvență maximă a ceasului de 16 MHz. Frecvența ceasului sistemului de dezvoltare 11,059200 MHz;
- memoria de date externă (DATA MEMORY), statică, implementată cu un circuit de tip UM62256AL, realizat în tehnologie CMOS, cu capacitatea de 32 kocteți şi caracterizat de un timp de acces de 35ns. Spațiul de adresare ocupat de memoria de date, în cadrul sistemului de dezvoltare, este cuprins între adresele 8000H şi FFFFH. Selectarea memoriei RAM se efectuează cu semnalele de control WR pentru scriere şi RD PSEN pentru citire.;

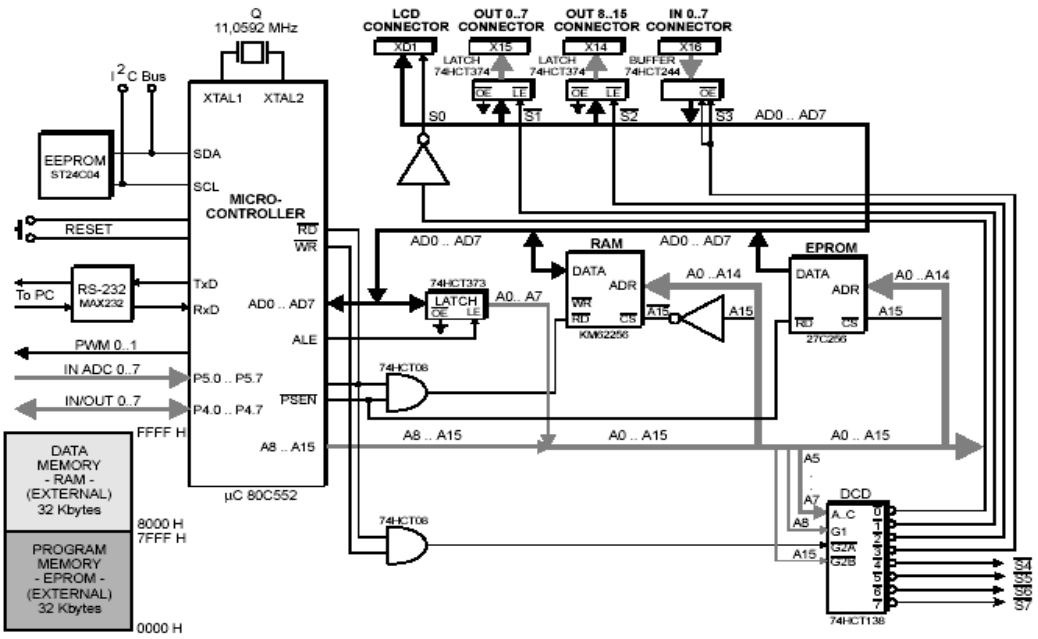


Figura 9.2. Schema sistemului de dezvoltare

- memoria de program externă (PROGRAM MEMORY), implementată cu un circuit EPROM de tip 27C256, realizat în tehnologie CMOS, cu capacitatea de 32 kocteți şi caracterizat de un timp de acces de 70ns. Spațiul de adrese ocupat de memoria de program externă în cadrul sistemului de dezvoltare este cuprins între 0000H şi 7FFFH. Selectarea memoriei RAM, activă pe nivel coborât, se efectuează cu semnalul A15, iar semnalul de control este PSEN pentru citire. Memoria externă de program conține programul de aplicație sau în faza de dezvoltare a acestuia conține un program monitor;
- interfață serială RS-232, full duplex, fără semnale de dialog, cu protocol software;
- bus serial I2C (bus multimaster cu arbitrare de priorități şi viteză mare de transmisie, frecvența maximă a ceasului serial este 100 kHz. Destinația principală este comunicația cu circuite integrate sau controller-e prevăzute cu interfața I2C;
- memorie EEPROM serială, implemetată cu un circuit de tip ST24C04, realizat în tehnologie CMOS, cu capacitatea de 512 octeți şi conectată la interfața I2C;
- 3 porturi paralele de 8 biți;
- 8 intrări multiplexate la un convertor analog-digital cu rezoluția de 10 biți, implementat în structura microcontroller-ului 80C552 şi caracterizat de un timp de conversie de 50 cicluri maşină (aproximativ 50 µs);

- 2 ieşiri analogice de 8 biţi modulate în durată. Prin integrarea lor se pot obține două convertoare digital-analogice de 8 biți;
- 3 numărătoare de tip timer / counter şi un watchdog programabil;
- 15 linii de întreruperi, dintre care 6 linii externe;
- 8 ieşiri decodificate de selecție porturi, specificate în cadrul tabelul 1:

Tabelul 1

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0		Adresa
0	X	X	X	X	X	X	1	0	0	0	X	X	X	X	X	S0	100h..11Fh
0	X	X	X	X	X	X	1	0	0	1	X	X	X	X	X	S1	120h..13Fh
0	X	X	X	X	X	X	1	0	1	0	X	X	X	X	X	S2	140h..15Fh
0	X	X	X	X	X	X	1	0	1	1	X	X	X	X	X	S3	160h..17Fh
0	X	X	X	X	X	X	1	0	0	0	X	X	X	X	X	S4	180h..19Fh
0	X	X	X	X	X	X	1	1	0	1	X	X	X	X	X	S5	1A0h..1BFh
0	X	X	X	X	X	X	1	1	1	0	X	X	X	X	X	S6	1C0h..1DFh
0	X	X	X	X	X	X	1	1	1	1	X	X	X	X	X	S7	1E0h..1FFh

Cele 8 semnale de decodificare pentru selectare porturi ocupă un spațiu de adrese cu dimensiunea de FFh, aşa după cum reiese din tabelul prezentat anterior. Dintre cele 8 semnale de selecție sintetizate, în structura sistemului de dezvoltare sunt utilizate doar 4, şi anume:

$\overline{S0}$ - semnal de selecție pentru afişajul cu cristale lichide LCD;

$\overline{S1}$ - semnal de selecție pentru portul de ieşire mai puțin semnificativ;

$\overline{S2}$ - semnal de selecție pentru portul de ieşire mai semnificativ;

$\overline{S3}$ - semnal de selecție pentru portul de intrare;

$\overline{S4} \div \overline{S7}$ - neutilizate (disponibile pentru extensii hardware);

Selectarea adreselor porturilor se face prin următoarea succesiune de instructiuni:

```
MOV P2,#1                          ; pentru A15÷A8
MOV R0,#(A7A6A5A4A3A2A1A0)B        ; pentru A7÷A0
MOV @R0,A                          ; în A se găseste valoarea trimisă
```

Extinderea numărului de porturi de intrare-ieşire poate fi făcută fie prin utilizarea semnalelor de selecție disponibile, ceea ce conduce la încărcarea magistralei interne a sistemului de dezvoltare, fie prin subdecodificarea liniilor inferioare de adrese neutilizate A4÷A0 şi multiplexarea, respectiv demultiplexarea, intrărilor, respectiv a ieşirilor, portului de intrare, respectiv a portului de ieşire mai puțin semnificativ.

Sub sistemul de operare Windows, ProView (Franklin Software) dispune de toate facilitățile software, adică: asamblor, compilator, program de conversie. Etapele parcurse pentru dezvoltarea unui program aplicație pe o platformă hardware coordonată de o unitate centrală de prelucrare cu microcontroller sunt următoarele:

- *Scrierea programului* cu ajutorul unui editor de texte. În sursa programului de aplicație sunt declarate explicit resursele suplimentare ale microcontrollerului 80C552 în raport cu 8051, într-o secțiune declarativă la început. Programul sursă de aplicație poate fi scris în limbajul de asamblare al familiei de microcontrollere 8051, caz în care numele său va fi PROGRAM.ASM sau într-o variantă specializată a limbajului C pentru familia de microcontrollere 8051, caz în care numele său va fi PROGRAM.C.
- *Asamblarea sursei* în urma căreia rezultă două fişiere:
  - PROGRAM.LST, un fişier listă ce este destinat localizării eventualelor erori rezultate în urma procesului de asamblare,
  - PROGRAM.OBJ, un fişier de tip obiect, care va fi folosit în continuare pentru obținerea formatului executabil;
- *Compilarea sursei* programului de aplicație;
- *Obținerea formatului executabil* de tip INTEL HEX, pe baza fişierului de tip OBJ;
- *Rularea programului* PROGRAM.HEX presupune rularea pe un calculator gazdă a unui program ce stabileşte o legătură la portul serial (Hyper Terminal), transferul fişierului Intel HEX şi apoi folosirea programului monitor inclus în EPROM-ul plăcii de dezvoltare. Programul se lansează în execuție cu comanda de monitor GO ADR, în care ADR reprezintă adresa la care este organizat programul în memoria microcontrollerului, specificată în prima linie a acestuia prin directiva ORG.

Standardul Intel HEX este unul din cele mai populare şi cele mai des folosite standarde din lumea 8051. Standardul este folosit pentru a introduce programul 8051 în EPROM, PROM, etc.; spre exemplu, un asamblor va genera un fişier Intel HEX care poate fi apoi încărcat într-un programator de EPROM şi apoi înscris în cip.

Un fişier Intel HEX este un fişier ASCI cu o singură “inregistrare” pe linie, fiecare linie are următorul format, tabelul 2:

Tabelul 2

Poziție	**Descriere**
1	**Marker de înregistrare:** primul caracter al liniei este întotdeauna “:” (ASCII 0x3A) pentru a identifica linia ca aparținând unui fisier Intel HEX

2 – 3	**Lungimea înregistrării:** acesta câmp conține numărul de octeți de dată, reprezentat ca un număr de 2 cifre hexazecimale. Acesta este numărul total de octeți de date, neincluzând octetul care reprezintă suma de control si nici primele 9 caractere ale liniei.
4 – 7	**Adresa:** acest câmp conține unde data ar trebui să fie încărcată în cip. Această valoare este de la 0 la 65.535 reprezentat ca un număr de 4 cifre hexazecimale.
8 – 9	**Tipul înregistrării:** acest câmp conține timpul de înregistrare din linia respectivă, valorile posibile sunt: **00** – înregistrarea conține data, **01** – End of File, **02** – adresă extinsă.
10 - ?	**Octeți de date:** următorii octeți sunt datele ce vor fi înscrise în EPROM. Data este reprezentată ca o valoare în hexazecimal de două cifre.
Ultimele 2 caractere	**Suma de control:** ultimele două caractere ale liniei sunt pentru suma de control. Suma de control reprezintă complementul sumei tuturor octeților de date predecesori excluzând octeții sumei de control şi caracterul ":" de la începutul liniei.

Exemplu:

:0300300002337A1E

lungime înregistrării: 03 (3 octeți de dată)
adresa: 0030 (cei 3 octeți vor fi stocați la adresele: 0030, 0031 şi 0032)
tipul înregistrării: 00 (data)
data: 02, 33, 7A
suma de control: 1E (03+00+30+00+02+33+7A=E2; 100-E2=1E)

Programul monitor este pornit odată cu punerea sub tensiune a plăcii şi este repornit de câte ori se apasă butonul de reset el facilitează transferul programelor în memoria RAM, depanarea programelor şi inițializează conexiunea serială. Setul de comenzi pot fi grupate în instrucțiuni:

- de afişare (DB, DC, DD,DI, DX, X), modificare (EB. EC, ED, EI, EX) şi completare (FILL)
- utilitare: asamblare(A), dezasamblare(U), execuție(G, T, P), stabilire şi folosire a punctelor de întrerupere (BD, BE, BK, BL, BS) şi încărcare/ salvarea memoriei de program.

Pentru a vedea o listă a tuturor comenzilor din meniu se poate tasta comanda *help* sau *?* în programul ce asigură legătura cu portul serial (Hyper Terminal) care va avea efectul din figura 9.3:

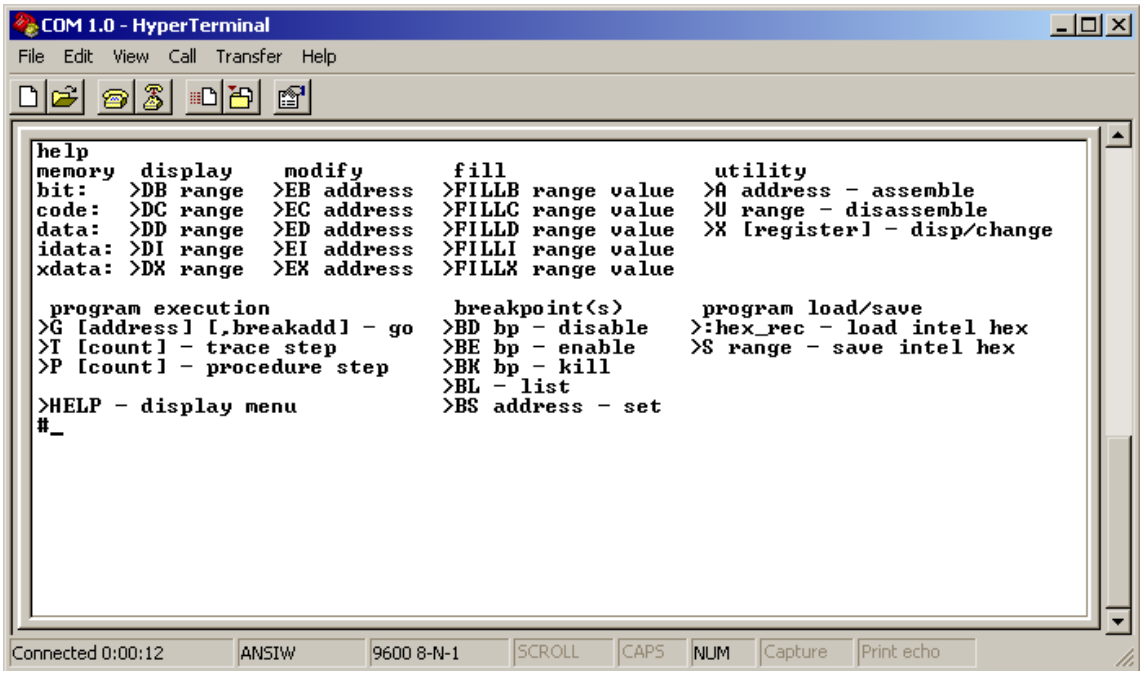


Figura 9.3. Programul monitor

Comenzile utile pentru depanarea programelor sunt: *trace step*, *procedure step*, cele pentru afişarea registrelor şi memoriei de program şi cele de stabilire a punctelor de ieşire din program. Deşi sunt utile, mult mai eficientă este utilizarea sistemelor de depanare disponibile în cadrul pachetului de programe ProView.

Procedurile de depanare pas cu pas rulează un număr variabil de instrucţiuni din programul aflat în memorie, la fiecare instrucţiune rulată se afişează conţinutul următoarelor registre: RA, RB, R0, R1, R2, R3, R4, R5, R6, R7, PSW, DPTR, SP şi PC.

Placa de dezvoltare se conectează la portul serial al calculatorului, legătura serială având viteza de 9600 biţi pe secundă, 8 biţi de dată, fără paritate, un bit de stop şi protocol Xon/Xoff.

1.2.Lucrare de familiarizare cu sistemul de dezvoltare

Scopul lucrării este familiarizarea cu sistemul, cunoaşterea setărilor ce trebuiesc făcute pentru a se putea comunica cu placa dar şi cunoaşterea modului de conectare a plăcii. Alimentarea plăcii sistemului de dezvoltare se face de la o sursă de alimentare de curent continuu sau de la sursa calculatorului. Legătura serială se face printr-un cablu ecranat cu trei fire (RX, TX şi GND) care are câte o cuplă de 9 pini la ambele capete. După conectarea plăcii la portul serial şi alimentarea plăcii se porneşte Hyper Terminalul. În urma lansării programului pe ecran trebuie să apară o fereastră ca în figura 9.4 unde trebuie dat un nume legăturii stabilite.

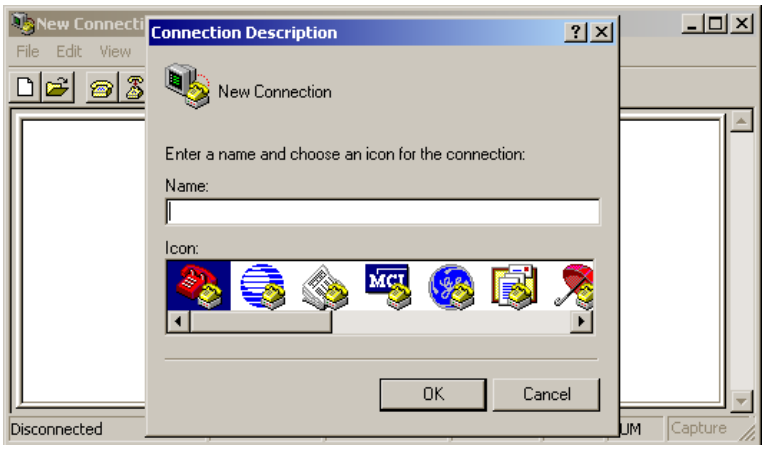


Figura 9.4. Hyperterminal

Pentru stabilirea legăturii se cere portul la care dorim conectarea şi proprietăţile legăturii, care se vor stabili ca în figura 9.5, ca să fie aceleaşi cu ale sistemului de dezvoltare. Pentru a vedea dacă stabilirea legăturii a fost realizată se apasă butonul de reset care va determina afişarea în zona de dialog a terminalului a prompterului monitorului, figura 5 dreapta.

Programele se încarcă în memoria de program cu ajutorul unor fişiere de tip Intel HEX, aceste fişiere fiind generate de către ProView. Modul în care programul se transferă către placă este ca şi pentru orice fişier text, deci din meniul terminalului se alege *Transfer* → *Send Text File*, se alege afişarea tuturor fişierelor după care se alege calea către fişierul generat de ProView, se selectează fişierul *.HEX şi se dă *Open*.

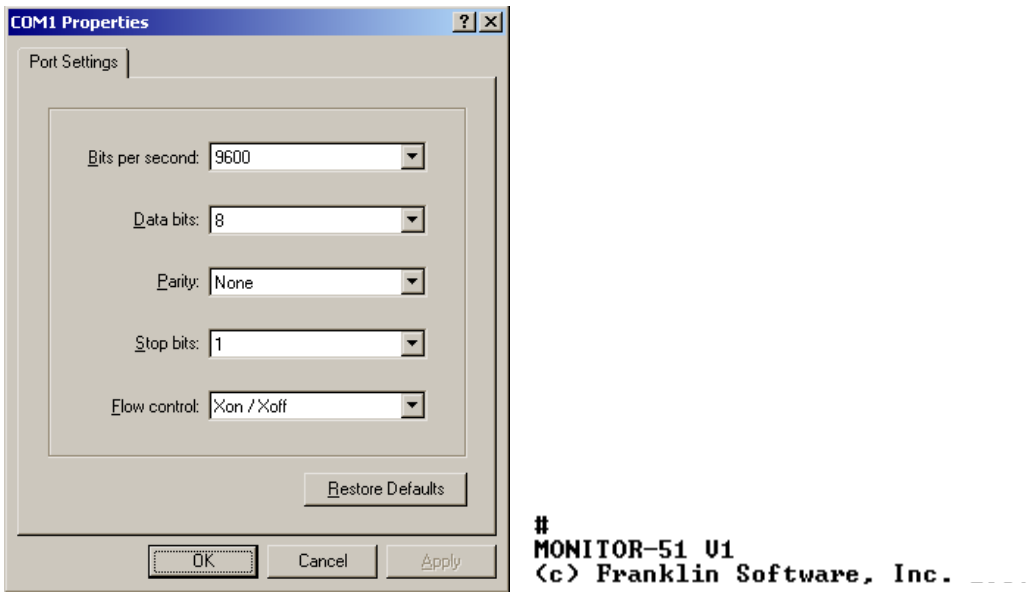


Figura 9.5. Setarea parametrilor transferului şi prompterul monitorului

Mediul de dezvoltare ProView produs de Franklin Software şi destinat dezvoltării de programe de aplicaţii pentru microcontrolere MCS 51 şi compatibile, pe lângă implementările compilatorului C51, asamblorului A51, editorului de legături L51 şi gestionarului de biblioteci LIB51, mai conţine:

- editor de texte sursă pentru fişiere multiple;
- manager de proiecte pentru crearea şi gestionarea fişierelor unei aplicaţii;
- ferestre de dialog pentru opţiunile mediului şi ale programelor de dezvoltare;
- sistem integrat de documentaţie on-line sensitiv la context;
- simulator/depanator simbolic, la nivel de cod sursă

După lansarea în execuţie a programului trebuie creat un proiect: *Project → New* se tastează numele proiectului, se alege calea directorului în care se lucrează şi se dă *OK*. Se pot crea fişiere de tip ASCI (*File → New*) cu extensia corespunzătoare limbajului în care se scrie programul (C sau limbaj de asamblare). Pentru ca aceste fişiere să poată fi compilate trebuie incluse în proiect: *Project → Add file*. Pentru ca la compilarea proiectului să se genereze şi fişierul Intel HEX trebuie setat acest lucru din cadrul opţiunilor linker-ului. Compilarea se face din cadrul meniului: *Project → Make* în urma acestei operaţii rezultă fişierul HEX.

Modul de simulare/depanare este pornit din meniu (*Debug → Start*), dispune de rularea automată sau pas cu pas (*Step into)*, evaluare/modificare expresii/simboluri ale aplicaţiei,

adăugare variabilă sau simbol de supravegheat. La depanare mai pot fi folosite şi optiunile din cadrul submeniului *View*, opțiuni care permit vizualizarea listingului, a regiştrilor principali de date şi SFR, a zonei de memorie selectate, a unor periferice (precum seriala), conținutul stivei, etc.

Vizualizarea face ca simularea şi depanarea să fie mult mai uşoare, de exemplu dacă in cadrul programului avem o instrucțiune *mov P0,#0Fh*, şi dacă se lansează *View* → *Hardware* → *Port 0*, se va vedea o fereastră precum cea din figura 9.6.

Exemplu de program:

```
ORG 8000h
nop
mov P0,#0Fh
nop
mov P0,#00h
end
```

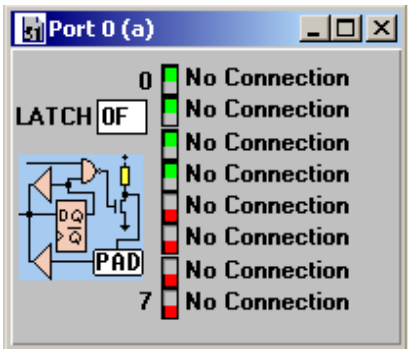


Figura 9.6. Program şi efectul rulării acestuia

Această succesiune de instrucțiuni, va trimite mai întâi la portul 0 valoarea 0x0F apoi următoarea instrucțiune mov va trimite la port valoarea 0, lucru foarte uşor de observat în fereastra portului 0.

Simularea şi depanarea se pot face şi pe placă, cu ajutorul programului monitor după transferul programului realizat. Se tastează T şi programul monitor va rula o instrucțiune după care va afişa modificările asupra regiştrilor principali. Se poate vizualiza memoria de date (DX), se pot stabili, activa şi dezactiva puncte de întrerupere a programului (BS, BE, BD).

Rezumat

Această lucrare începe ciclul celor 3 lucrări dedicate studiului interfațării cu microcontrollere. Lucrarea face o prezentare generală a sistemului de dezvoltare format din placa IMC 500 şi mediul de dezvoltare software Franklin. Este prezentată selecția pe magistrala externă a microcontrollerului pentru a sublinia legătura cu partea de teorie.
În lucrare se prezintă modul de lucru care constă în scrierea programului, verificarea lui şi înscrierea în microcontroller, apoi rularea pe microcontroller şi eventual depanarea.

Bibliografie

1. C. Gerigan, P. Ogrutan, *Tehnici de interfațare*, Editura Universitatii Transilvania Brasov, 2000, pag. 201-229 online la: http://vega.unitbv.ro/~ogrutan/ti/cap10.pdf
2. http://www.fsinc.com/devtools/index.htm
3. http://www.nxp.com/documents/data_sheet/80C552_83C552.pdf
4. http://www.keil.com/dd/docs/datashts/intel/ism51.pdf