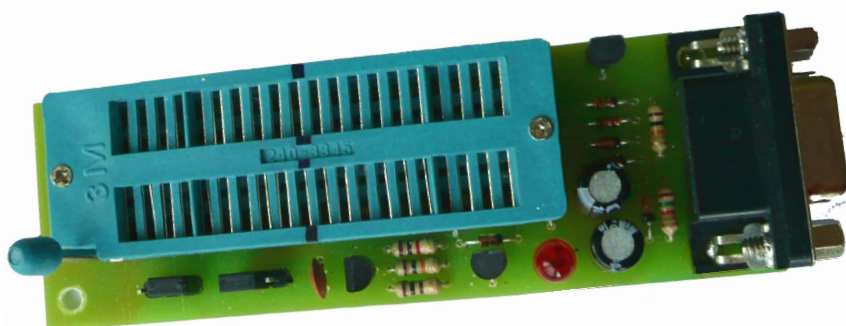


Cuprins

Fișa de Asamblare	
1. Funcționare	2
2. Schema	3
4. Lista de componente	3
5. PCB	4
6. Introducere în μ Controllere	5



PROGRAMATOR PIC

- Avantaj Pret/Calitate
- Livrare rapida
- Design Industrial
- Proiecte Modificabile
- Adaptabile cu alte module
- Module usor de asamblat
- Idei Interesante

Idei pentru afaceri

Hobby & Proiecte Educationale

Foarte popular, acest programator este destinat familiei de microcontrollere cu 8 – 40 pini ale firmei MICROCHIP, folosind un singur soclu ZIF .

Lista controllerelor

- PIC12C508,PIC12C509
- PIC12C508A,PIC12C509A
- PIC12CE518,PIC12CE519
- PIC12C671,PIC12C672,PIC12CE673,PIC12CE674
- PIC12F508,PIC12F509

- PIC12F629,PIC12F635(1)(2),PIC12F675,PIC12F683(2)
- PIC16C505
- 16C62B(3),
- PIC16F627,PIC16F628
- PIC16F627A(2),PIC16F628A,PIC16F648A
- PIC16F630,PIC16F636(2),PIC16F676,PIC16F684(2),PIC16F688(2)
- ,PIC16C711(3),
- PIC16C716(3)
- PIC16C745(3),PIC16C765(3)
- PIC16C771(7)
- PIC16C773,PIC16C774
- PIC16C781(7),
- PIC16F818,PIC16F819
- PIC16F84
- PIC16F84A
- PIC16F87(2),PIC16F88
- PIC16F870(8),PIC16F871(8),PIC16F872,PIC16F873,PIC16F874(4),PIC16F876(5),PIC16F877(5)
- PIC16F873A(5),PIC16F874A(5),PIC16F876A,PIC16F877A
- PIC18F1320,PIC18F2320,PIC18F4320
- PIC18F242,PIC18F252,PIC18F442,PIC18F452
- PIC18F248,PIC18F258,PIC18F448,PIC18F458(3)
- PIC18F4539(3)

Funcționare

Se alimentează direct din portul serial, așadar trebuie să verificăm dacă tensiunile TxD , DTR , RTS la portul serial al calculatorului PC-ului sunt mai mari de +7.5 V (sau - 7.5V), pentru ca programatorul să funcționeze bine, întrucât noile laptop-uri folosesc circuite integrate interfațată RS232 cu consum redus de energie ce nu au capacitatea de a alimenta programatorul nostru. Chiar și așa, putem să îl alimentăm extern, fiind mult prea bun pentru a nu-l folosi. Ceea ce trebuie să știm:

1 . Linia de masă GND și de alimentare VDD sunt specifice schemei programatorului și nu se conectează la masa portului serial sau la linia de alimentare fără modificările necesare.

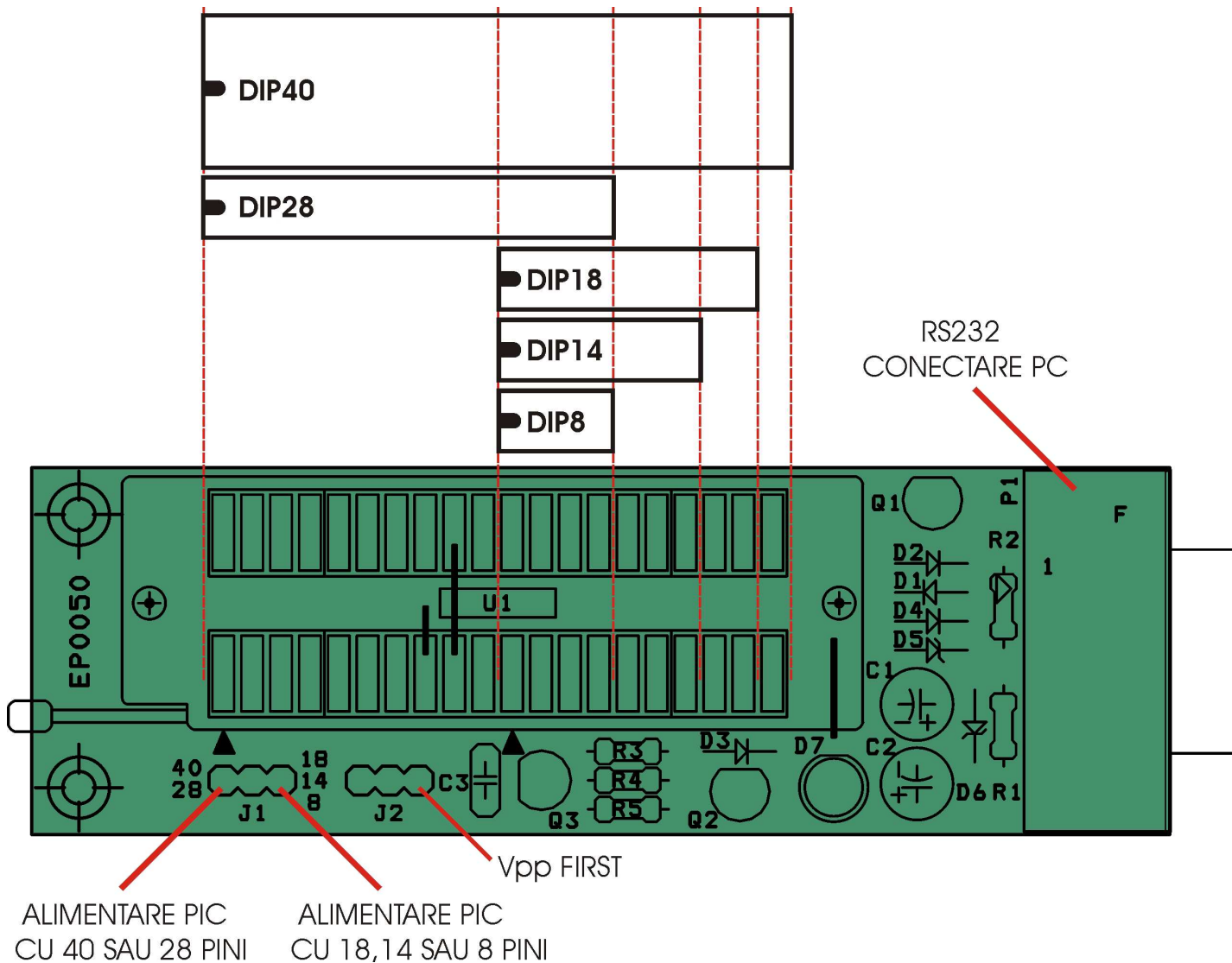
2 . Acest programator schimbă alimentarea VPP în

conformitate cu dispozitivul selectat (8 - 18Pini) sau (28 - 40pini) , din jumperi . Deci, în cazul nu se setează corect jumperul pentru tipul de capsulă, riscul de a distruge controllerul este foarte mare;

3 . Acest programator nu suportă toate PIC-urile. Prin utilizarea unor adaptoare însă, acestea pot fi programate (de ex. 16C770/771)

4 Software-ul folosit pentru programare este **IC - Prog** de la Bonny Gijzen sau **winpic**

5 . " Setările hardware " ale IC - Prog sunt aceleași ca la programatorul JDM.



Amplasarea componentelor

Bibliografie
Multi PIC Programmer 5 Ver.2
WinPicProg 1.91

Acest produs se livrează în varianta circuit imprimat, circuit imprimat + componente sau în varianta asamblată în scopuri educaționale și va fi însoțit de documentația completă de asamblare pe CD.

Dacă doriți să aflați mai multe despre produsele noastre, vizitați situl www.epsicom.com

Dacă ați întâmpinat probleme cu oricare dintre produsele noastre sau dacă doriți informații suplimentare, contactați-ne prin e-mail office@epsicom.com

Pentru orice întrebări, comentarii sau propuneri de afaceri nu ezitați să ne contactați pe adresa office@epsicom.com

31 Sararilor Street | 200570 Craiova, Dolj, Romania | 0723.377.426, 0743.377.426

Câteva cuvinte despre structura microcontrollerelor ?

Cu ajutorul acestor prime informații veți intra în superba lume a microcontrollerelor.

În primul rând să știm cu ce lucrăm, cum funcționează și cum putem realiza programe pentru ele.

Câteva linii de program reușite și câteva exemple revelatoare vă vor da impulsul la a transforma un chip într-o jucărie teribilă. Ce ne trebuie pentru început ? Curajul abordării, în fond nu este complicat deloc ! Totul are un început iar apoi vine și performanța.

Schema bloc simplificată

La PIC16F84 sau PIC16F870 vom găsi și vom lucra cu acest grup de patru blocuri.

Memoria program (verde) va conține programul pe care îl înscrinem. Programul (Software-ul) conține un set de date și instrucțiuni pe care microcontrollerul le va prelucra. Ele vor fi scrise într-un program cu ajutorul unui PC, instrucțiuni pe care le înscrinem apoi în controller, adică le programăm cu un programator în "memoria program".

Această memorie este de tip EEPROM care poate fi rescrisă de mii de ori. Cum o înscrinem ? Cu un programator pe care îl veți putea găsi în colecția IT ([EP0050](#) de exemplu pentru o gamă largă de microcontrolere produse de firma MICROCHIP).

Regiștrii și blocul RAM (portocaliu) conține toate registrele interne precum și o mică memorie RAM (64-128 octeți) unde puteți stoca date temporare. Există mai mulți regiștrii cu funcții diferite ce vor fi explicați în continuare.

Pentru ce sunt necesari ? Să luăm un exemplu: dacă faci o buclă program, atunci ai nevoie de o variabilă pentru a schimba valoarea de fiecare dată când bucla se execută și apoi folosim o variabilă definită într-o adresă RAM pentru a păstra valoarea contorului. (Vom reveni mai târziu cu un exemplu edificator, nu este complicat). Conținutul registrului și datele din RAM vor dispărea la oprirea tensiunii de alimentare, sunt doar date provizorii, temporare.

O altă zonă de memorie, care va funcționa la fel ca și RAM-ul, este memoria **EEPROM (galben)** Aceasta este o memorie mică unde puteți citi precum și scrie date, însă datele nu vor dispărea atunci când tensiunea dispăre. Datele din această memorie pot fi adresate, citite, modificate sau descărcate, însă pot fi și protejate la citire cu un cod (lock).

Ultimul bloc este cel cu **porturi**, adică pinii de intrare sau ieșire pe care îi definim prin program ca intrări sau ieșiri cu funcțiile pe care le dorim (să citim anumite date prin pinii-port de intrare și să oferim datele rezultate din program către anumite circuite cum ar fi de comandă unor elemente de forță, ...). În exemplele ce vor urma afla cum și la ce se folosesc.

PIC16F870 este o versiune apărută ulterior procesorului PIC16F84, ambele circuite conțin mai multe blocuri decât cele descrise. Trecând de la prezentarea generală trecem la citirea fișelor tehnice ale celor două circuite pentru a obține detaliile fiecărui bloc, pentru a le utiliza numai dacă avem strict nevoie de ele. Ambele controlere au timere, watchdogs, sistem de întreruperi și multe alte blocuri.

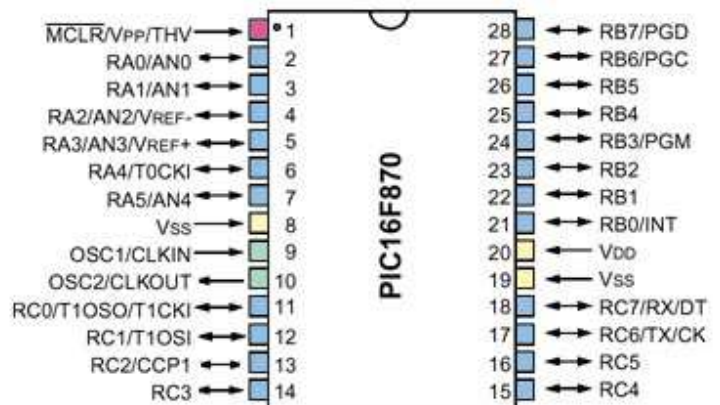
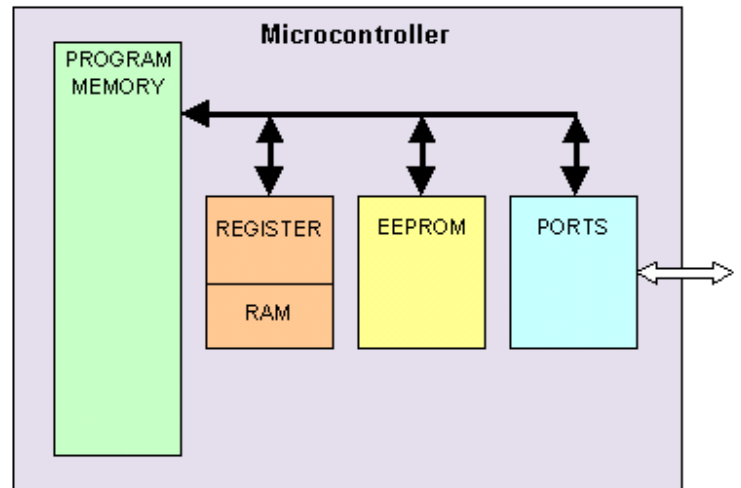
Să luăm un exemplu: circuitul PIC16F870

În figura alăturată sunt notati pinii circuitului.

Pentru o mai bună selecție pe categorii a acestora, s-au colorat.

Astfel:

- Pinii desenați cu albastru sunt porturi (pini de intrare/ieșire). Deoarece nu există multe extra funcții în acest microcontroller, unii pini pot fi folosiți în mai multe scopuri: ca intrări la un convertor AD (analog/numeric) intern iar alți pini pot fi conectați la un contor intern, etc. Prima oară vom afla cum utilizăm pinii albaştrii ca simple intrări și ieșiri.
- Cei doi pini verzi ar trebui să fie conectați la un cuarț pentru a obține un semnal de ceas intern, generatorul de impulsuri de tact.
- Pinii galbeni sunt dedicați alimentării circuitului;
- Pinul roșu este intrarea Reset care va inițializa funcționarea circuitului.



Până aici a fost relativ simplu de înțeles.

Poate părea uneori greu și plictisitor, însă observațiile din exemple ne scapă mai tot timpul de explicarea detaliilor.

Vrem să învățăm să programăm și nu știm cu ce să începem? De ce anume mai avem nevoie pentru a ne pune la treabă ?

Etapa I

- În primul rând avem nevoie de o platformă de evaluare/dezvoltare a unor circuite realizate cu microcontrollere.
- Luăm un exemplu de circuit din manual sau de pe net, configurăm circuitul și testăm programele deja realizate. Exemplele sunt un mod rapid de învățare. Deschidem fișierele ASM, citim comentariile făcute pentru fiecare linie și vom înțelege foarte repede modul în care funcționează microcontrolerul, vom învăța de asemenea un set minim de instrucțiuni. Foile de catalog ale fiecărui procesor conțin detalii referitoare la instrucțiuni.

Etapa II

Iată o carte EBOOK de bază despre PIC16F84.

[Aici găsim o documentație excelentă pentru înțelegerea funcționării și elemente de programare pentru PIC16F84.](#)

Merită citită, este gratuită.

Ce putem face, de exemplu, pentru a aprinde un LED în primă fază și să îl facem apoi să clipească ?

- vom scrie câteva linii de program și le vom salva într-un fișier cu extensia .asm (**[le găsim în cartea de mai sus](#)**). Pentru o înțelegere rapidă a procedurii și o corectare rapidă, liniile de program din .asm sunt și vor fi însoțite de comentarii.
- îl asamblăm cu un program numit MPASM (vezi mai jos),
- îl compilăm rezultând un fișier .hex în cod mașină și un fișier .lst de verificare (erorile apărute în limbajul de asamblare pe linia de program vor fi regăsite ca mesaje în fișierul .lst). Fișierul .hex va fi înscris în format recunoscut de procesor.
- Verificăm liniile de program și apoi funcționarea aplicației programând controllerul (vezi mai jos).

Etapa III

Ce mai rămâne de făcut ?

Dacă ați ajuns la acest rând înseamnă că ați făcut deja foarte multe. De aici înainte folosiți imaginația pentru a realiza aplicații cât mai interesante. Mult succes cu proiectele voastre și nu uitați că cei mai buni profesioniști au fost autodidacții !

Asamblorul

Există multe asambleoare pentru PIC16F84 și 16F870 însă ce poate fi mai la îndemână decât programul oferit gratuit de producătorul **MICROCHIP**, ultima versiune de **MPLAB** ? Microchip a realizat un mediu software complet unde puteți simula software-ul vostru înainte de a realiza circuitul. Acest mediu este numit MPLAB. **Development software (MPLAB® IDE)**

Programatorul

Dacă ați asamblat codul veți obține un fișier hex. pe care va trebui să îl înscriseți în microcontroller, adică să programați microcontrollerul.

Pentru a face acest lucru avem nevoie de o interfață (Programator PIC) de la computer la microcontroler și de un program software cu care să transferăm fișierul din PC în microcontroller.

Iată un programator util pentru o gamă largă de microcontrollere din seria MICROCHIP, cod **EP0050** realizat și de firma noastră. După instalarea programului și cuplarea programatorului pe portul serial, se lansează programul, se selectează controllerul PIC16F870 și se setează opțiunile de programare (configurare):

Fuserele:

- Watchdogtimer (pagina 101 datasheet)
- Poweruptimer (pagina 89 datasheet)
- Brown out resetare permite (pagina 94 datasheet)
- Low Voltage Programming Enable (Activare tensiune mica de programare)
- Code Protection Data Enable (Activare Codul de protecție a datelor)
- Flash Program Memory Write Enable
- Brown out reset enable (pagina 103 datasheets)

Fișierele necesare de la Microcip: **[PIC16F870 Datasheet](#)**, **[Programarea specifică pentru PIC16F87X](#)**