

### Cuprins

Fișa de Asamblare	
1. Funcționare	2
2. Schema	3
4. Lista de componente	3
5. PCB	4
6. Introducere în $\mu$ Controlere	5

## PLATFORMĂ DE EVALUARE 2 $\mu$ CONTROLLERE

- 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

**Construcție On Board, numit și Single-Board computer pentru mult prea cunoscutul microcontroller PIC16F84, este poate cea mai comună placă cu utilizare directă, uneori finală, la aplicațiile noastre.**

### Caracteristici:

Construcție all OnBoard  
Alimentare 7,5-16Vcc  
Oscilator 4 – 20 MHz  
Extensie EEPROM  
Stabilire praguri intrări analogice  
LCD Display alfanumeric  
10 Led-uri  
16 taste  
ISP  
Comunicație RS232  
Comunicație RS485  
Transmisie IR  
Speaker  
Conector RJ12  
Conector Servo  
Acces direct la porturi  
Zonă de lucru suplimentară

Board Complet cu clock de 4MHz, 8MHz, 10MHz, 12MHz, 13.32MHz, 16MHz sau 20MHz.

Compatibil cu procesoare cu 40 pini:

**PIC16C64(A), 16C65(B), 16C662, 16C67, 16C74(AB), 16C765, 16C77, 16C774, 16F74, 16F747, 16F77, 16F777, 16F871, 16F874, 16F874A, 16F877, 16F877A, 16F914, 16F917, 18C442, 18C452, 18F4220, 18F4320, 18F4331, 18F4410, 18F442, 18F4420, 18F4431, 18F4439, 18F4455, 18F448, 18F4480, 18F4510, 18F4515, 18F452, 18F4520, 18F4525, 18F4539, 18F4550, 18F458, 18F4580, 18F4585, 18F4610, 18F4620, 18F4680** , recomandat **16F877A**

### Funcționare

Pe acest board se poate testa orice program pe arhitectură realizată cu PIC16F84, cu ceasul de 4 - 20MHz.

**Clock** - cu comutatorul K2 se selectează regimul: ceas cu quartz sau ceas realizat cu un oscilator RC la care frecvența se ajustează din P1.

**Reset** -cu switch-butonul S3 direct pe MCLR

**Programarea** – in-circuit prin schimbarea tensiunii pe MCLR la 13.5V cu ajutorul unui circuit TL497 tip step-up convertor. Regimul este ales cu S1 și semnalizat cu D21.

**RS232** - conectarea serială cu PC-ul se face printr-un conector DB9 **numai în regim off** (nealimentat). Semnalele DATA și CLOCK sunt trimise pe RB6 și RB7 prin driverele IC4C și d iar prin IC4b și a se transmite către interfață semnalul de pe RB7, respectiv reset și regim lucru/programare.

**Afisaj Led-uri**, starile RA și RB sunt afisate cu led-urile D8-D20. Afisarea starilor sunt de un real folos la depanarea programelor la frecvente mici. Regimul de afisare este stabilit cu S8.

**Tastatura** - cu trei push-butoane S5, S6 și S7 se modifică starea intrărilor RA1 și RA0 (separat sau împreună prin SAU

cablat) permițând setarea sistemului într-un mod similar cu setarea unui ceas (regim, up-down).

**Afișarea alfanumerică** cu LCD 2x16 caractere este controlată cu semnalele E (validare) RS (selecție registru) conectate la RA2 și RA3. Pinul R/W al afișajului este conectat la masă, regim scriere. Cu S9 validăm afișarea pe LCD iar cu S4 și P2 avem controlul luminii și contrastului.

Alimentarea se face de la o sursă externă de 9-12V prin K1 și este semnalizată cu D22. Cu stabilizatorul IC1 obținem tensiunea de 5V.

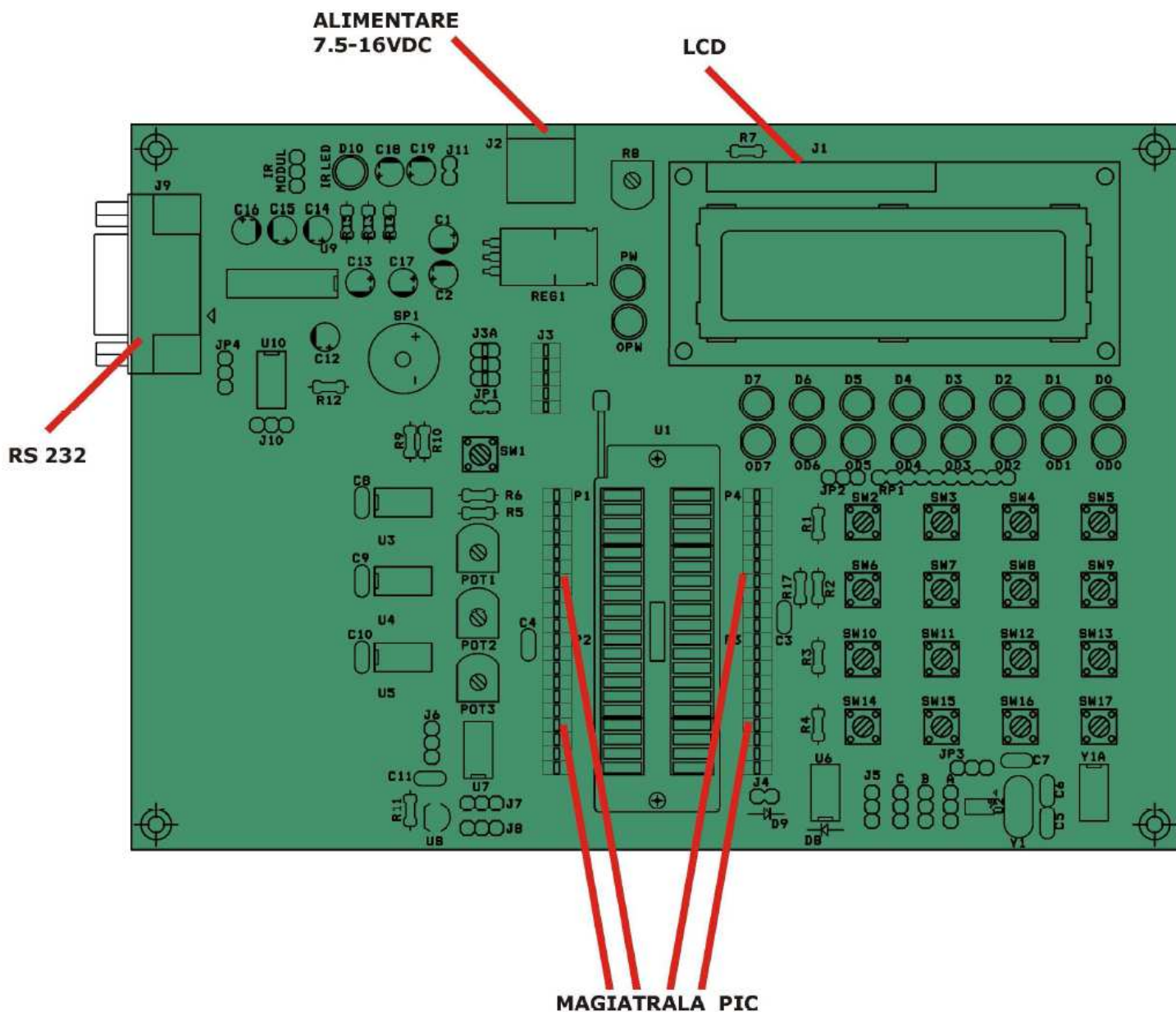
### Programe

Sunt necesare câteva programe pentru inițiere și utilizare acestui sistem (programe de test pe care le găsim rapid pe internet). Astfel MPLAB.EXE (editor, assembler și simulator) și MPASM.EXE (assembler) și PSIM.EXE (simulator) sunt downloadabile free de pe internet, **NTPicprog**, de la adresa <http://tmw.dreamhosters.com/ntpicprog.zip> , **ICPROG.EXE** de la [www.ic-prog.com](http://www.ic-prog.com) e super. Alte resurse de programare ?



## Lista de componente

Nr.Crt.	Componenta	Denumire	Valoare	Cant
1	JP2,JP3,JP4,C,B,A	Conector	CON3	6
2	C1,C17	Condensator polarizat	10 $\mu$ F	2
3	C2,C12,C13,C14,C15,C16	Condensator polarizat	1 $\mu$ F	6
4	C3,C4,C8,C9,C10,C11, C18, C19	Condensator nepolarizat	100nF	8
5	C5,C6	Condensator nepolarizat	22pF	2
6	C7		10nF	1
7	D1,0D1,D2,0D2,D3,0D3,D4, 0D4,D5,0D5,D6,0D6,D7,0D7, PW,D0,0PW,0D0	LED	LED	18
8	D8,D9	Diodă	1N4148	2
9	D10	Diodă IR	IR LED	1
10	IR MODULE	Receptor IR	IR IN	1
11	JP1,J11	Conector	CON2	2
12	J1	Display	LCD	1
13	J2	Conector	Jack	1
14	J3	Conector	Con 2x5	1
15	J3A	Conector	Con 2x3	1
16	J4	Conector	Con2	1
17	J5	Cuarț	32,768KHz	1
18	J6, J7, J8, J10	Conector	Con3	4
19	J9		DB9 MAMA	1
20	POT1,POT2,POT3		5K $\Omega$	3
21	P1,P2,P3,P4	Soclu șir	2x10	4
22	REG1	C.I.	LM7805T	1
23	RP1	Rezistență SIL	470 $\Omega$ x9	1
24	R1,R2,R3,R4,R15	Rezistență	270 $\Omega$	5
25	R5,R7,R13,R14	Rezistență	1K $\Omega$	4
26	R6	Rezistență	100 $\Omega$	1
27	R8	Semireglabil	10K $\Omega$	1
28	R9,R10,R11	Rezistență	4,7K $\Omega$	3
29	R12	Rezistență	120 $\Omega$	1
30	R17	Rezistență	100K $\Omega$	1
31	SP1	Buzzer	HY-05	1
32	SW1,SW2,SW3,SW4,SW5,SW6,SW7, SW8,SW9,SW10,SW11,SW12, SW13,SW14,SW15,SW16,SW17	Push Buton	SW KEY-SPST	17
33	U1	Soclu Ziff	PIC16F877	1
34	U2	C.I.	ICS502M	1
35	U3	C.I.	I2C SEEPROM	1
36	U4	C.I.	SPI SEEPROM	1
37	U5	C.I.	Microwire SEEPROM	1
38	U6	C.I.	NJU6355	1
39	U7, U8	C.I.	DS1620	2
40	U9	C.I.	MAX232	1
41	U10	C.I.	SN75176A	1
42	Y1A, Y1	Quarț	4,000 MHz	2



Amplasarea componentelor

#### Bibliografie

1. Multi PIC Programmer 5 Ver.2
2. WinPicProg 1.91
3. <http://www.microengineeringlabs.com>
4. <http://www.techsystemembedded.com>
5. <http://www.microchip.com>
6. <http://melabs.com>

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](http://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](mailto:office@epsicom.com)

Pentru orice întrebări, comentarii sau propuneri de afaceri nu ezitați să ne contactați pe adresa [office@epsicom.com](mailto: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 înscrim. Programul (Software-ul) conține un set de date și instrucțiuni pe care microcontrolerul le va prelucra. Ele vor fi scrise într-un program cu ajutorul unui PC, instrucțiuni pe care le înscrim 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 înscrim? 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.

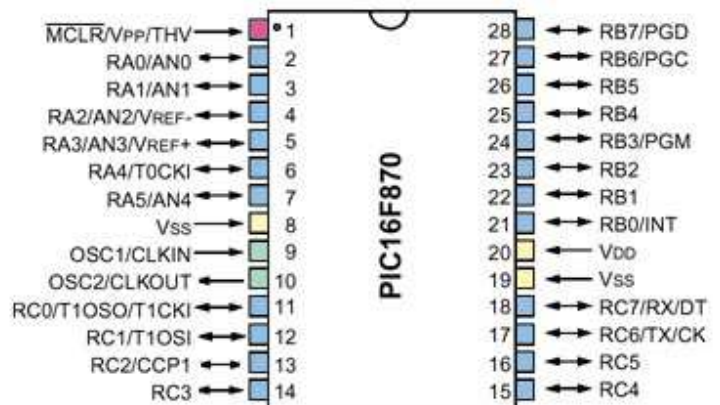
Astfel:

- 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.
- Verificați 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 înregistrați 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.

Dupa instalarea programului și cuplarea programatorului pe portul serial, se lansează programul, se selectează controllerul PIC16F870 și se setează opțiunile de programare (configurare):

Fuserile:

- 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](#)