

Cuprins

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

SUPER PROBE - V3

- 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

Pornind de la proiectul de bază al lui Luhan Monat s-a dezvoltat acest proiect interesant în jurul unui PIC16F870 câteva componente, un afișaj pe 4 digiți și ... un soft genial. Date despre proiect le găsiți pe site-ul autorului <http://mondo-technology.com/super.html>

Un proiect realizat cu PIC16F870, un laborator întreg într-un ... super tester.

Funcționare

Schema cuprinde procesorul ce folosește un ceas pe quart, afișajul KW-561ASA de 9mm/digit și câteva componente , a căror rol va fi descris la regimurile de lucru.

Soft-ul este astfel conceput încât se aprinde pe rând câte un segment (multiplexare/segment) eliminând astfel efectul de pâlpâire la consum variabil. Se folosesc componente cu rol bine definit, astfel, R5 este folosit de generator, R4 la măsurarea capacităților.

Funcțiile sunt selectate prin apăsarea celor două push butoane (buton 1 validare, buton 2 selectare funcție).

Prob – Sonda Logică (Logic Probe). În acest regim arată 'H' pentru 1 (peste 3.7 volti), 'L' pentru 0 (sub 0.8 volti) și '-' pentru intermediar. Dacă este detectat un impuls (minimum 0.5 μ s), va apare 'P' cu intermitențe în locația a doua;

PULS – Generatorul de impulsuri (Logic Pulser) Când butonul 1 este apăsat, se generează o serie de impulsuri de 0.5 μ s iar pe afișaj se aprinde mijlocul segmentului. La apăsarea butonului 2 în ultimele locații se afișează frecvența (5,50,500, 5.0). Rata de impulsuri este memorată și după oprirea alimentării;

FrEq – Frecvențmetru (Frequency Counter), mod selectat cu butonul 1, se afișează frecvența în KHz (prin apăsare primele 4 cifre la prin ridicare ultimele cifre). Dacă punctul zecimal apare intermitent frecvența este afișată în MHz;

Cnt – Contor (Event Counter). În acest regim se afișează ultimii patru digiți cu butonul 1 ridicat iar la apăsare primii patru digiti. Cu butonul 2 se șterge contorul;

VoLt – Voltmetru , cu o eroare de 2%, tensiunea de alimentare fiind referință.

**NU MĂSURAȚI ÎN NICI UN CAZ
TENSIUNI PESTE 5 VOLȚI!**

diod – Măsurarea tensiunii de joncțiune (Diode Junction Voltage), afișare directă ca și la voltmetru, curentul prin joncțiune printr-o rezistență de 10K Ω ;

Cap – Capacimetru (Capacitance Measurement), Valori cuprinse între 0.001-500 μ F. Pentru valori mari (sute de μ F) timpul de măsură durează câteva secunde;

SIG – Generator de Semnal . Prin apăsarea butonului 2 se generează un semnal de 500Hz la 0.5V;

ntSC – Generator de Miră NTSC (Video Patern) , imagine cu puncte albe;

Generator de caractere ASCII (9600 - Serial Ascii) – la apăsarea butonului 1 se generează codul ASCII al caracterelor de la A la Z urmat de cr/lf. Cu butonul 2 se setează rata de 1200, 2400, 4800 sau 9600 baud;

Generator Servo (R/C Servo) de la 1ms la 2ms pentru acționări servo. Cu butonul 1 se mărește iar cu butonul 2 se scade durata. Valoarea implicită 1.5ms;

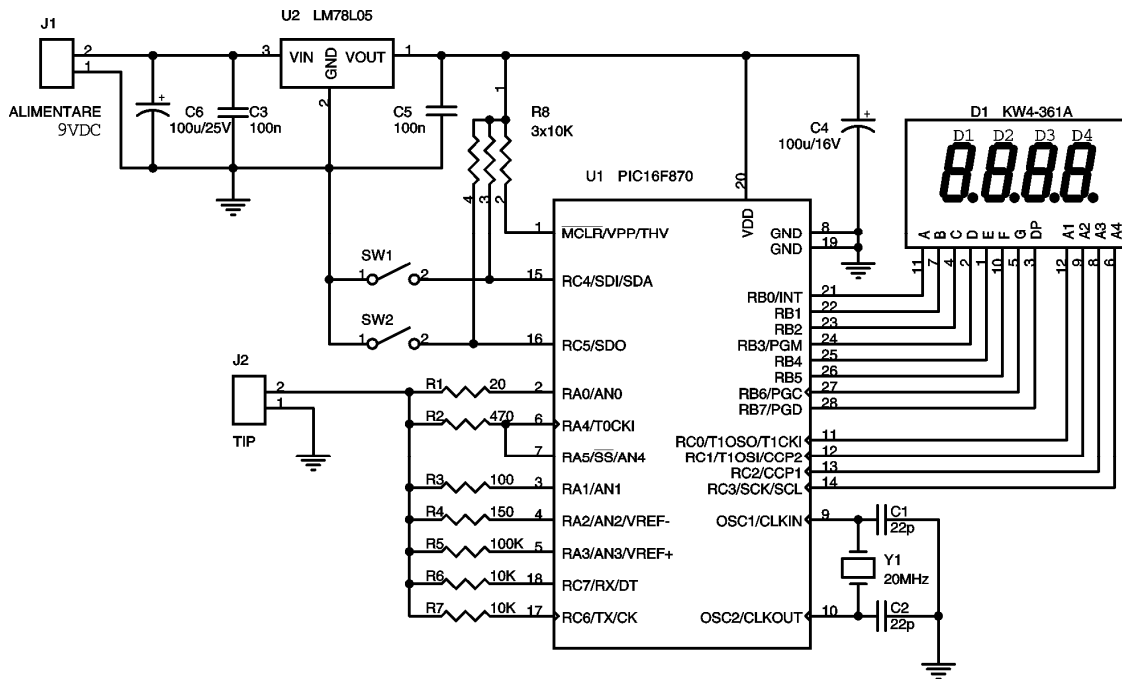
[] – Square waves – Generează frecvențe 1-9999 Hz. Prin butonul 1 se crește frecvența iar prin butonul 2 scădem frecvența;

Prn – Generator de numere aleatoare (Pseudo Random Number) la frecvența de 10KHz;

ir38 – Cod IR – Generează impulsuri de 1 și 2 ms la 38KHz pentru testarea receptorului de IR;

PWM – semnale variabile de la 3-97% la 6KHz. Cu butonul 1 și 2 se crește sau se descrește factorul de umplere. În orice mod, apăsând cele două butoane se trece în Menu.

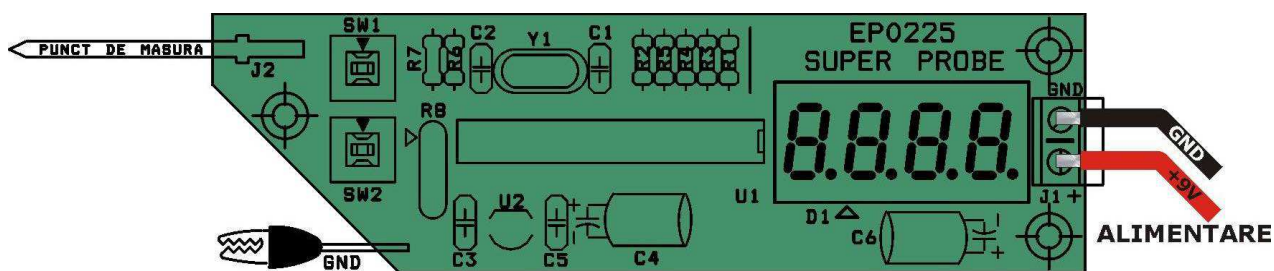
Prin apăsarea butonului 1 se selectează modul iar cu butonul 2 se revine în modul anterior. La alimentare se trece în regimul ultim selectat (este memorată starea).



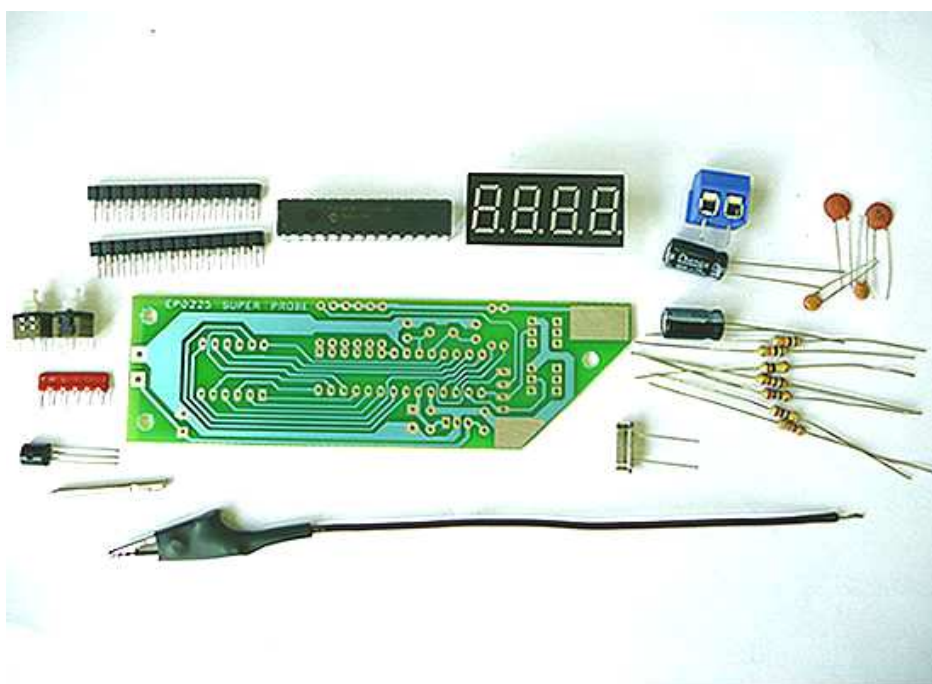
Schema electrică

Nr.Crt.	Componenta	Denumire	Valoare	Cant
1	C2,C1	Condensator	22pF	2
2	C5,C3	Condensator	100nF	2
3	C4	Condensator	100µF/16V	1
4	C6	Condensator	100µF/25V	1
5	D1	Afişaj LED	KW-561ASA	1
6	J1, J2	Conector	CON2	1
7	R1	Rezistență	20Ω	1
8	R2	Rezistență	470Ω	1
9	R3	Rezistență	100Ω	1
10	R4	Rezistență	150Ω	1
11	R5	Rezistență	100KΩ	1
12	R6,R7	Rezistență	10KΩ	2
13	R8	Rezistență şir	3x10KΩ	1
14	SW1,SW2	Switch	SW KEY-SPST	2
15	U1	C.I.	PIC16F870+soclu	1
16	U2	C.I.	LM78L05	1
17	Y1	Cuartz	20MHz	1
18	J2	Conector	PIN+CROCODIL	1

Lista de componente



Amplasarea componentelor



Bibliografie

MONDO TECHNOLOGY - programul sursă și date despre autor

Acest produs se livrează în varianta asamblată sau în varianta circuit imprimat + componente 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).

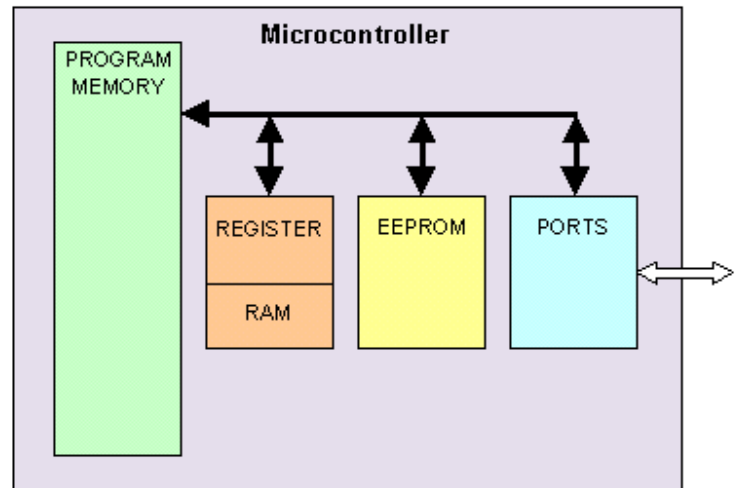
Regiștrii și blocul RAM (portocaliu) conține toate registrele interne precum și o mică memorie RAM (64-128 octeți) unde puteți stoca date temporare. Există mai mulți regiștrii cu funcții diferite ce vor fi explicați în continuare.

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 pastra 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 aceasta 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.



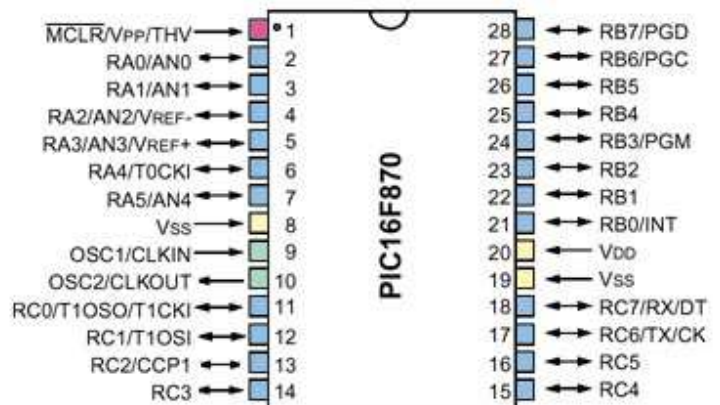
Să luăm un exemplu: circuitul PIC16F870

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 alții 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 asambleatoare 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](#)**