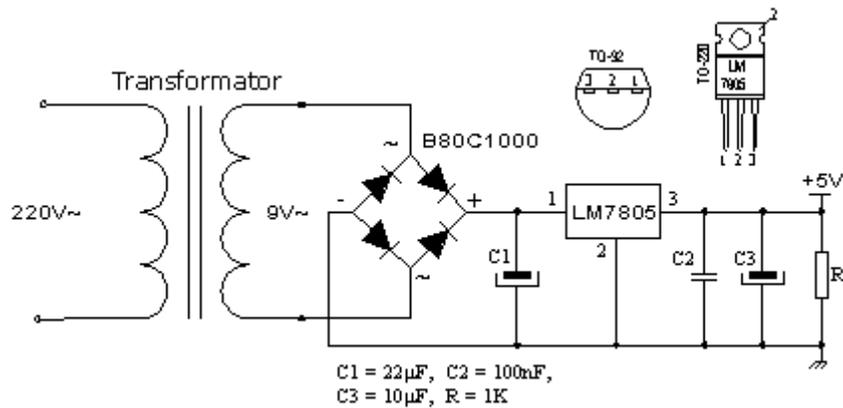


Introducere

Exemplele oferite în această capitole vă vor arăta cum să conectați microcontrolerul PIC cu alte componente sau dispozitive periferice când produceți propriul sistem bazat pe microcontroler. Fiecare exemplu conține descriere detaliată a părții hardware cu schema electrică și comentarii despre program.

Alimentarea microcontrolerului

În general, alimentarea corectă este de o importanță maximă pentru funcționarea corectă a sistemului cu microcontroler. Poate fi ușor comparată cu respirația unui om în aer. Este mai probabil ca un om care respiră în aer curat să trăiască mai mult decât un om care locuiește într-un mediu poluat. Pentru o funcționare corectă a oricărui microcontroler, este necesar să oferim o sursă stabilă de alimentare, un reset sigur în momentul în care îl porniți și un oscilator. Conform specificațiilor tehnice oferite de producătorul microcontrolerului PIC, tensiunea de alimentare ar trebui să se încadreze între 2.0V și 6.0V pentru toate versiunile. Cea mai simplă soluție este folosirea stabilizatorului de tensiune LM7805 care oferă tensiune stabilă de +5V la ieșire. O astfel de sursă este ilustrată în figura de mai jos.



Pentru a funcționa corect sau pentru a avea o tensiune stabilizată la 5V la ieșire (pinul 3), tensiunea de intrare pe pinul 1 la LM7805 ar trebui să fie între 7V și 24V. În funcție de curentul consumat de montaj vom folosi tipul corespunzător de stabilizator de tensiune LM7805. Sunt diferite versiuni de LM7805. Pentru consum de curent de până la un 1A ar trebui să folosim versiunea în capsulă TO-220 cu posibilitatea de răcire adițională. Dacă consumul total este de 50mA, putem să folosim 78L05 (versiune de stabilizator în capsulă mică TO-92 pentru curent de până la 100mA).

Dacă folosim un oscilator (rezonator) de 4MHz, pentru valorile prescaler-ului 0,1 și 7 care divid ceasul de bază al oscilatorului, intervalul urmat de o depășire a contorului TMR0 va fi 0.512, 1.02 și 65.3ms. Practic, aceasta înseamnă că cea mai mare întârziere va fi 256x65.3ms care este egală cu 16.72 secunde.

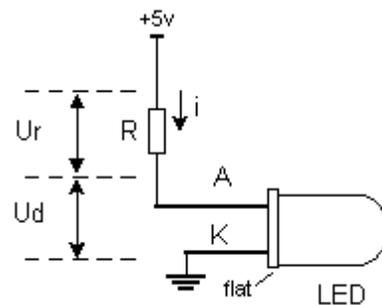
Prescalerul	Divizor	Depășire
b'00000000'	1:2	0.512 ms
b'00000001'	1:4	1.02 ms
b'00000111'	1:256	65.3 ms

Example

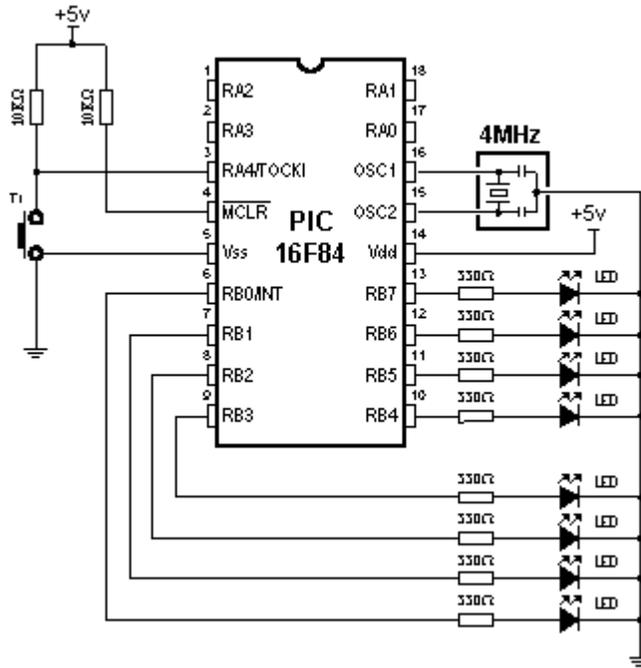
Light Emitting Diodes –LEDuri

Ledurile sunt unele dintre cele mai folosite elemente în electronică. LED este o abreviere pentru „Light Emitting Diode”. În momentul în care alegem un led, sunt mai mulți parametri de care trebuie să ținem seama: diametrul, care este de obicei 3 sau 5mm (milimetri), curentul de funcționare care este în jur de 10mA (poate fi mai mic decât 2mA pentru ledurile cu randament maxim: emisie de lumină puternică) și bineînțeles culoarea, care poate fi roșie sau verde deși mai sunt leduri portocalii, albastre, galbene... . Ledurile trebuie conectate corect pentru a emite lumină și rezistența care limitează curentul trebuie să fie de o valoare corectă pentru ca ledul să nu se ardă (supraîncălzire). Tensiunea pozitivă de alimentare este legată la ANOD, iar catodul este legat la tensiunea negativă sau la masa circuitului. Pentru a identifica fiecare pin, catodul este cel mai scurt pin iar corpul are în general o teșitură pe partea catodului. Diodele vor emite lumină numai dacă curentul circulă de la ANOD spre CATOD. Altfel jonctiunea PN este polarizată invers și curentul nu va circula. Pentru a conecta corect un led trebuie adăugată o rezistență în serie pentru a limita de curentul prin diodă, pentru ca aceasta să nu se ardă. Valoarea rezistenței este determinată de curentul care vreți să circule prin led. Curentul maxim care poate **curge** printr-un led a fost stabilit de producător. Ledurile cu randament maxim pot produce rezultate bune cu un curent mai mic de de 2mA.

Pentru a determina valoarea rezistenței serie, trebuie să cunoaștem valoarea tensiunii de alimentare. De aici scădem tensiunea care cade pe led. Această valoare va varia de la 1,2v la 1,6v, depinzând de culoarea ledului. Răspunsul este valoarea lui U_r . Folosind această valoare și curentul care vrem să circule prin LED (între 0.002A și 0.01A) putem să aflăm valoarea rezistenței cu ajutorul formulei: $R = U_r / I$.



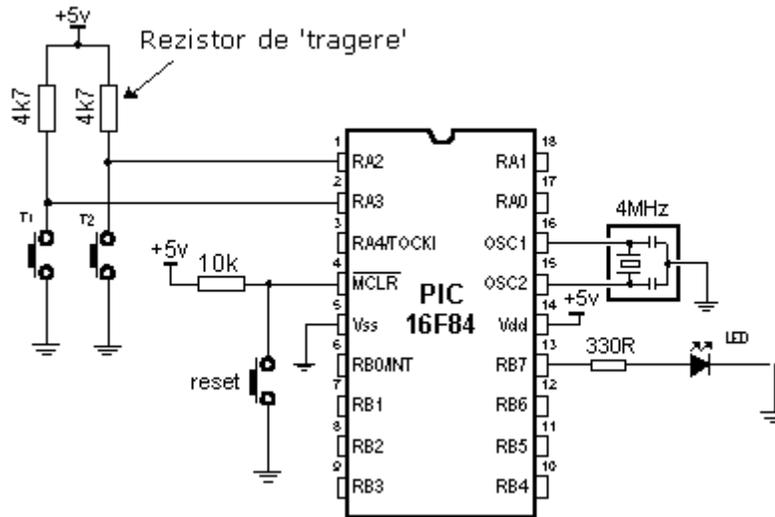
Ledurile sunt conectate la microcontroler în două metode. Una este să le activăm cu zero logic și a doua este să le activăm cu unu logic. Prima metodă este numită logică NEGATIVĂ iar cea de-a doua este numită logică POZITIVĂ. Figura de mai sus ilustrează modalitatea de conectare prin logică POZITIVĂ. Deoarece logica POZITIVĂ oferă o tensiune de +5v diodei și rezistenței serie, ledul va emite lumină de fiecare dată când un pin al portului B este în starea 1 logic (1 = ieșire HIGH). Logica NEGATIVĂ necesită ca ledul să fie întors și terminalele de tip anod să fie conectate împreună la borna pozitivă a sursei. În momentul în care este livrată o ieșire LOW de la microcontroler către anod și rezistență, ledul va lumina.



Connecting LED diodes to PORTB microcontroller

Tastatura

Tastaturile sunt dispozitive mecanice utilizate pentru a executa o întrerupere sau pentru a realiza o conexiune între două puncte. Ele au diferite mărimi și au diferite scopuri. Tastele care sunt utilizate aici sunt denumite „taste dip”. Ele sunt lipite direct pe o placă de circuit și sunt deseori întâlnite în electronică. Au patru pini (doi pentru **fiecare contact**), ceea ce le oferă stabilitate mecanică.



Exemplu pentru conectarea tastelor la pinii microcontrolerului

Funcția tastei este simplă. În momentul în care apăsăm o tastă, două contacte sunt unite și se realizează o conexiune. Totuși, nu toate lucrurile sunt simple. Problema constă în natura tensiunii ca valoare, și în imperfecțiunea contactelor mecanice. Înainte ca un contact să fie realizat sau decuplat, există o perioadă scurtă de timp când pot apărea vibrații (oscilații) ca rezultat al imperfecțiunii contactelor mecanice, sau din cauza vitezei diferite de apăsare (acest lucru depinde de persoana care apasă tasta). Termenul atribuit acestui fenomen este denumit **switch (contact) debounce**. Dacă acest lucru nu este prevăzut în momentul în care un program este conceput, poate apărea o eroare sau programul poate produce mai mult decât un singur impuls la ieșire pentru o singură apăsare de tastă. Pentru a evita acest lucru, putem introduce o mică întârziere când detectăm închiderea unui contact. Aceasta va asigura faptul că apăsarea unei taste este interpretată ca un singur impuls. Întârzierea de **debounce** este produsă în software și durata întârzierii depinde de buton și de scopul butonului. Problema poate fi parțial rezolvată prin adăugarea unui condensator în paralel la tastă, dar un program bine realizat oferă rezultate mai bune. Programul poate fi ajustat până când detecția falsă este complet eliminată. În anumite cazuri o simplă întârziere poate fi suficientă dar dacă vrei ca programul să se ocupe de mai multe lucruri în același timp, o simplă întârziere va însemna că procesorul nu va face nimic pe o lungă perioadă de timp și poate rata alte intrări sau poate decupla portul de ieșire către un afișor. Soluția este să avem un program care să urmărească apăsarea unei taste cât și decuplarea unei taste. Macro-ul de mai jos poate fi folosit pentru **keypress debounce**.

Optocuplor

Optocuplorul combină un LED și un fototranzistor în aceeași capsulă. Rolul unui optocuplor este acela de a separa două părți de circuit.

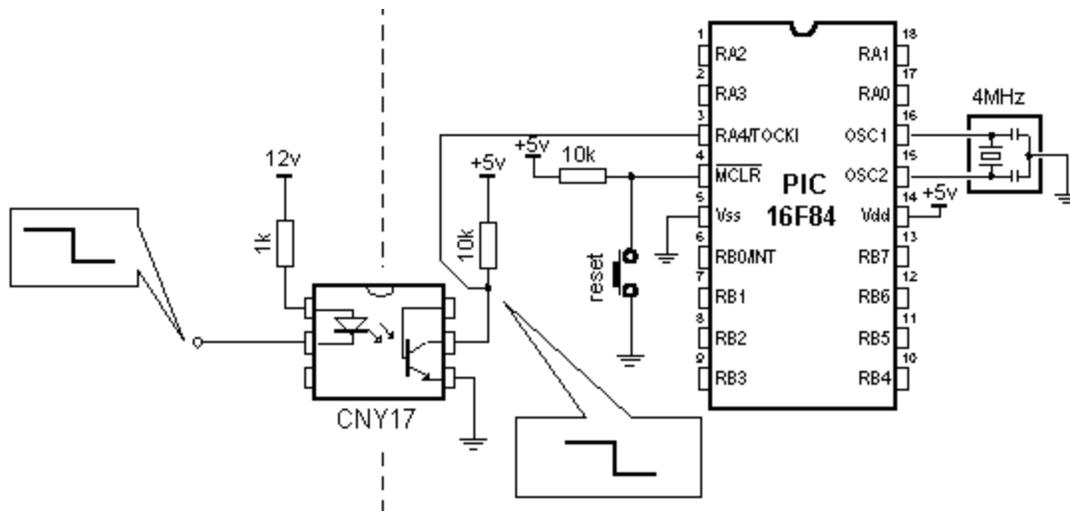
Aceasta este realizată pentru un număr de motive:

- **Interferența.** O parte a unui circuit poate fi într-o zonă unde este influențat de interferențe (cum ar fi cele de la motoarele electrice, echipamente de sudură, motoare termice etc.). Dacă ieșirea acestui circuit trece printr-un optocuplor spre alt circuit, numai semnalele dorite vor trece prin optocuplor. Semnalele de interferență nu vor avea destulă „putere” să activeze LEDul din optocuplor și de aceea ele sunt eliminate. Exemplele tipice sunt unitățile industriale care au mai multe interferențe care afectează semnalele pe cablu. Dacă aceste interferențe afectează funcția unei secțiuni de control, vor apărea erori și unitatea nu va mai funcționa.
- **Separare simultană și intensitatea semnalului.** Un semnal mai mic de 3v este capabil să activeze un optocuplor și ieșirea optocuplorului poate fi conectată la o linie de intrare a microcontrolerului. Microcontrolerul are nevoie de un impuls de intrare de 5v și în caz semnalul de 3v este amplificat la 5v. Poate fi folosit pentru a amplifica curentul semnalului. Uitați-vă mai jos pentru utilizarea unei linii de ieșire a microcontrolerului pentru amplificare de curent.
- **Separare de tensiune mare.** Optocuploarele au calități înnăscute pentru separarea tensiunilor mari. Deoarece LEDul este complet separat de fototranzistor, optocuploarele pot da dovadă de izolare de tensiune de 3Kv sau chiar mai mare.

Optocuploarele pot fi folosite ca dispozitive de intrare sau ieșire. Ele au funcții adiționale cum ar fi Schmitt triggering (ieșirea unui Schmitt trigger este 0 sau 1 – se schimbă încet ridicând și coborând forma de undă în valori definite LOW sau HIGH). Optocuploarele sunt împachetate ca o singură unitate sau în grupuri de două sau mai multe într-o singură capsulă. Ele mai sunt denumite foto-întrerupătoare în care un disc cu fante este introdus într-un lăcaș între LED și fototranzistor și de fiecare dată când lumina este întreruptă, tranzistorul produce un impuls. Fiecare optocuplor are nevoie de două alimentări pentru a funcționa. Ele pot fi folosite cu o alimentare, dar capacitatea de izolare a tensiunii este pierdută.

Optocuplor pe o linie de intrare

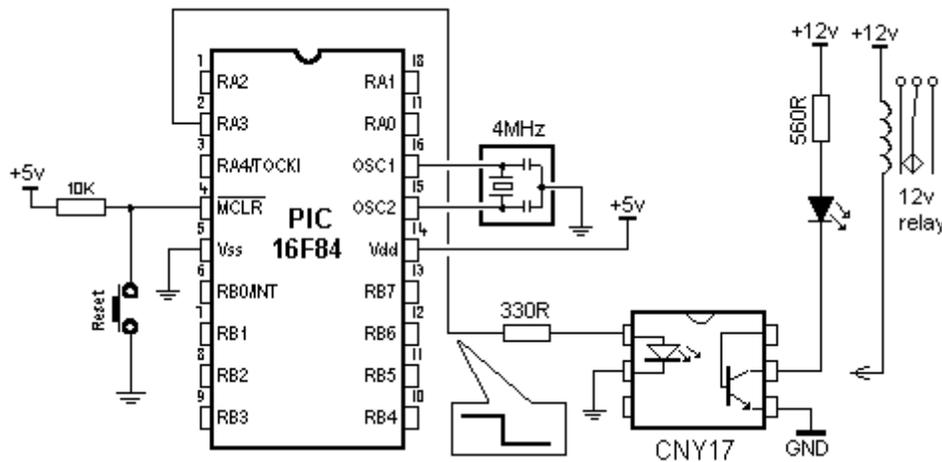
Modul de funcționare este simplu: când ajunge un semnal, LEDul din optocuplor este aprins și luminează pe baza fototranzistorului din aceeași carcasă. În momentul în care tranzistorul este activat, tensiunea dintre colector și emitor cade la 0.5v sau mai puțin și microcontrolerul sesizează acest lucru ca zero logic pe pinul RA4. Exemplul de mai jos este un contor, folosit pentru numărarea produselor de pe o linie de producție, pentru determinarea vitezei motorului, pentru contorizarea numărului de revoluții a unei axe etc. Considerăm senzorul ca un microîntrerupător. De fiecare dată când întrerupătorul este închis, LEDul este luminat. LEDul „transferă” semnalul către fototranzistor și operația fototranzistorului livrează LOW către intrarea RA4 a microcontrolerului. Un program în microcontroler va fi necesar pentru a preveni contorizările false și un indicator conectat la oricare dintre ieșirile microcontrolerului va indica starea curentă a contorului.



Exemplu de linie de intrare cu optocuplor

Optocuplor pe o linie de ieșire

Un optocuplor poate fi folosit pentru a separa semnalul de ieșire a unui microcontroler față de un dispozitiv de ieșire. Acest lucru poate fi necesar pentru separarea tensiunilor înalte sau pentru amplificare. Ieșirea unor anumite microcontrolere este limitată la 25mA. Optocuplorul va lua semnal de curent scăzut din microcontroler și tranzistorul de ieșire va comanda un LED sau un relee, cum este exemplificat mai jos:

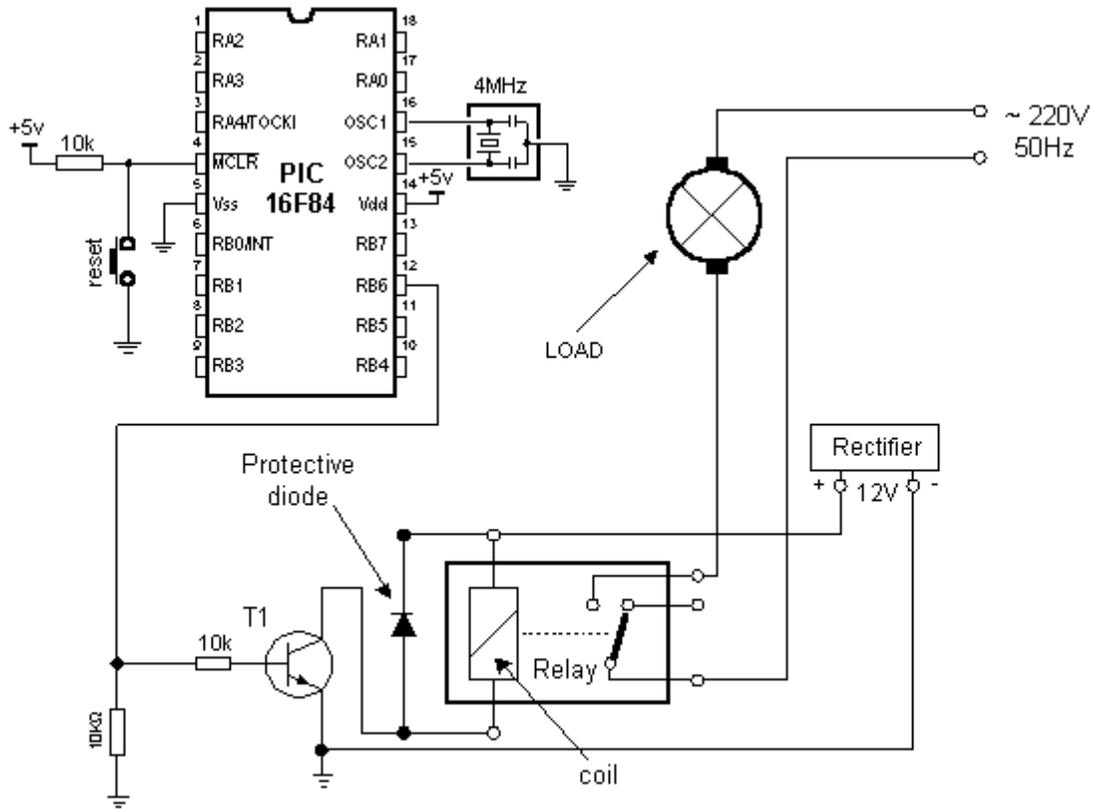


Output line optocoupler example

Programul pentru acest exemplu este simplu. Prin livrarea unui '1' logic în pinul 4 al portului A, LEDul se va aprinde și tranzistorul va fi activat în optocuplor. Orice dispozitiv conectat la ieșirea optocuplorului va fi activat. Curentul limită pentru tranzistor este în jur de 250mA.

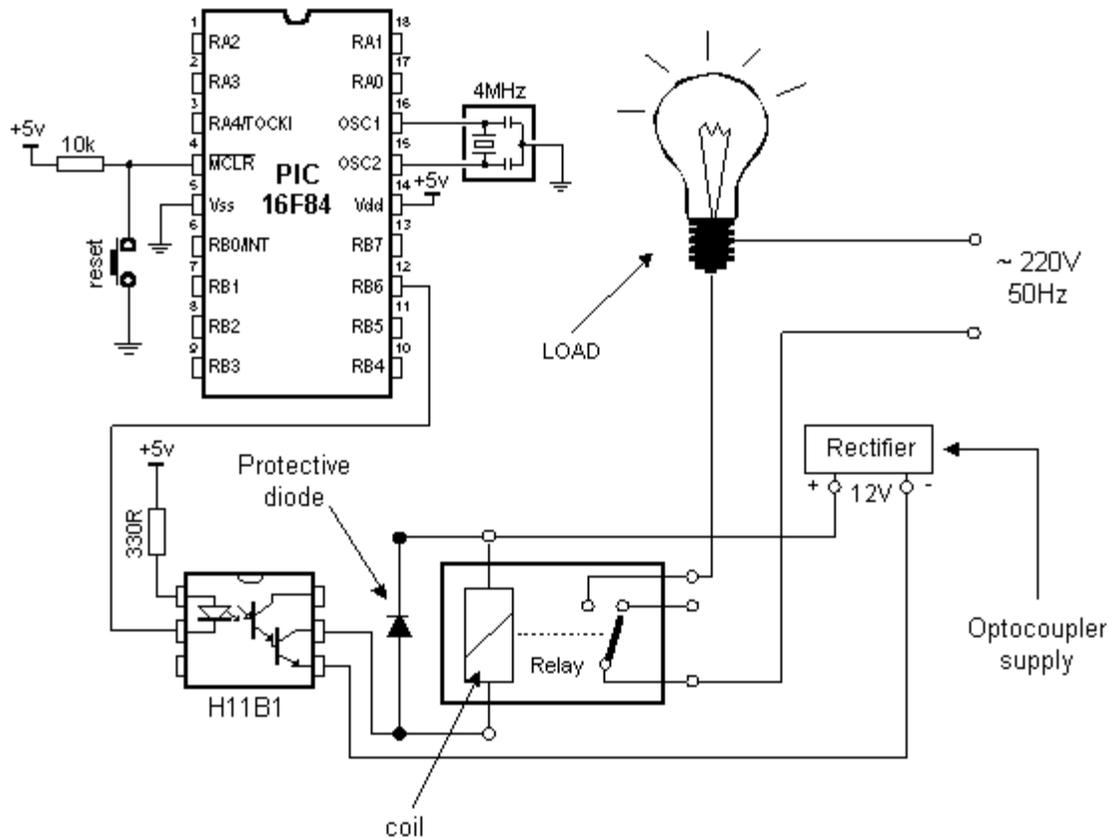
Releul

Releul este un dispozitiv electromecanic care transformă un semnal electric într-o mișcare mecanică. El este alcătuit dintr-o bobină din conductori izolați înfășurați pe un nucleu metalic și o armătură metalică cu unul sau mai multe contacte. În momentul în care o tensiune de alimentare este aplicată la bornele unei bobine, curentul circulă și va fi produs un câmp magnetic care mișcă armătura pentru a închide un set de contacte și/sau pentru a deschide un alt set. Când alimentarea este dezactivată din releu, cade fluxul magnetic din bobină și se produce o tensiune înaltă în direcția opusă. Această tensiune poate strica tranzistorul de comandă și de aceea este conectată o diodă cu polarizare inversă de-a lungul bobinei pentru a scurtcircuita vârfurile de tensiune în momentul în care apar.



Conectarea unui releu la microcontroler prin intermediul unui tranzistor

Multe microcontrolere nu pot comanda un releu direct și de aceea un tranzistor de comandă este necesar. Un HIGH pe baza tranzistorului activează tranzistorul și acesta la rândul lui activează releul. Releul poate fi conectat la orice dispozitiv electric prin intermediul contactelor. Rezistența de 10K din baza tranzistorului limitează curentul dinspre microcontroler la o valoare solicitată de tranzistor. Rezistența de 10K dinspre bază și bara negativă previne ca tensiunile de zgomot aplicate în baza tranzistorului să activeze releul. De aceea numai un semnal clar de la microcontroler va activa releul.



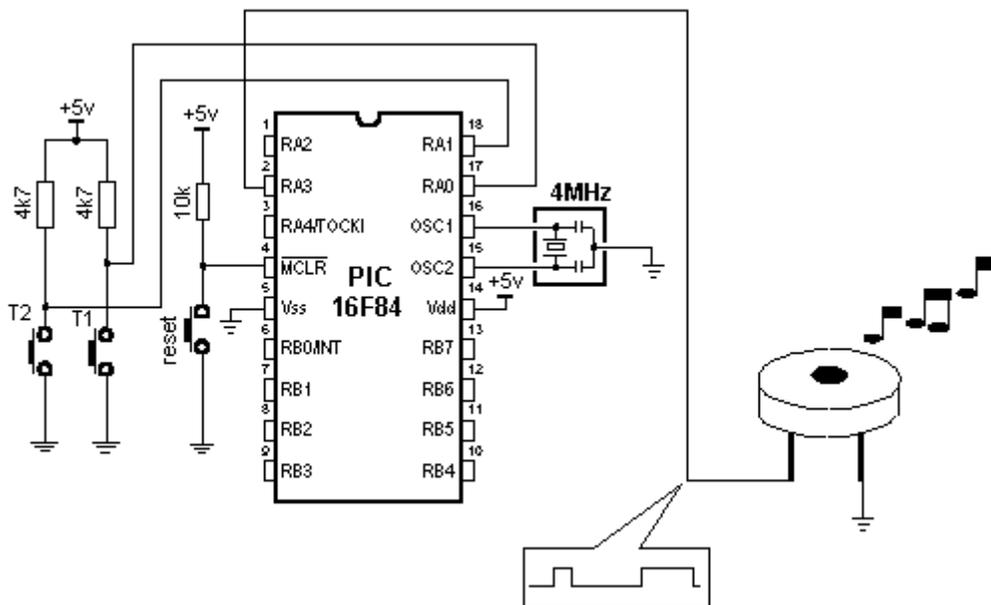
Connecting the optocoupler and relay to a microcontroller

Un releu poate fi de altfel activat prin intermediul unui optocuplor care în același timp amplifică curentul provenit de la ieșirea microcontrolerului și oferă un grad înalt de izolare. Optocuploarele **HIGH CURRENT** de obicei conțin un tranzistor cu o ieșire „Darlington” pentru a oferi curent mare de ieșire. Conectarea prin intermediul unui optocuplor este recomandată în mod special pentru aplicațiile microcontroler unde motoarele sunt activate și zgomoturile de comutație provenite de la motor pot ajunge în microcontroler prin intermediul liniilor de alimentare. Optocuplorul comandă un releu iar releul activează motorul. Figura de mai jos arată programul necesar pentru activarea releului și include câteva din macrourele deja discutate.

Generarea unui sunet

Un buzzer piezo poate fi adăugată la o linie de ieșire a unui microcontroler pentru a livra tonuri „audio”, piuituri și semnale. Este important de știut că sunt două mari tipuri de dispozitive piezoelectrice emițătoare de sunet. Una are componente active înăuntrul carcasei și are nevoie numai de alimentare de curent continuu pentru a emite un ton sau un beep. În general tonurile sau beep-urile emise de aceste difuzoare sau piuitoare nu pot fi schimbate – ele sunt fixe din cauza circuitelor interne. Acesta nu este tipul despre care discutăm în acest articol. Celălalt tip constă dintr-un buzzer piezo și necesită semnal livrat în ea pentru a funcționa. Depinzând de frecvența formei de undă, ieșirea poate fi ton, melodie, alarmă sau chiar un mesaj vocal. Pentru ca ele să funcționeze trebuie să livrăm un ciclu care este alcătuit din semnale HIGH și LOW. Tranziția de la HIGH la LOW sau de la LOW la HIGH cauzează mișcări diafragmei pentru a

produce secvențe de sunete. Forma de undă poate avea o schimbare fină de la o valoare la alta (denumită undă sinusoidală) sau o schimbare rapidă (denumită undă dreptunghiulară). Un calculator este ideal pentru producerea de unde dreptunghiulare. Livrarea de unde dreptunghiulare produce o ieșire ușor grosieră. Conectarea unui buzzer piezo este foarte ușoară. Un pin este conectat la linia negativă, iar cealaltă la o ieșire a microcontrolerului, după cum este ilustrat în figura de mai jos. Acesta va livra o formă de undă de 5v către buzzerul piezo. Pentru a produce o tensiune mai mare, forma de undă trebuie amplificată și aceasta necesită un tranzistor de comandă și o bobină.



Conectarea unui buzzer piezo la un microcontroler

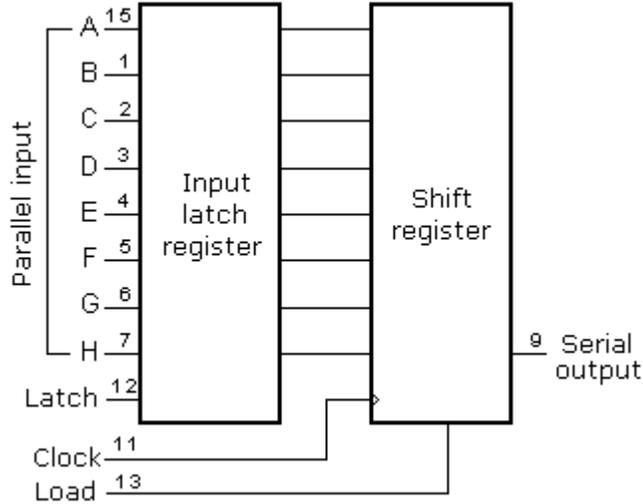
Ca și în cazul tastaturii, puteți folosi un macro care va furniza o rutină BEEP într-un program când va fi necesar.

Regiștrii de deplasare

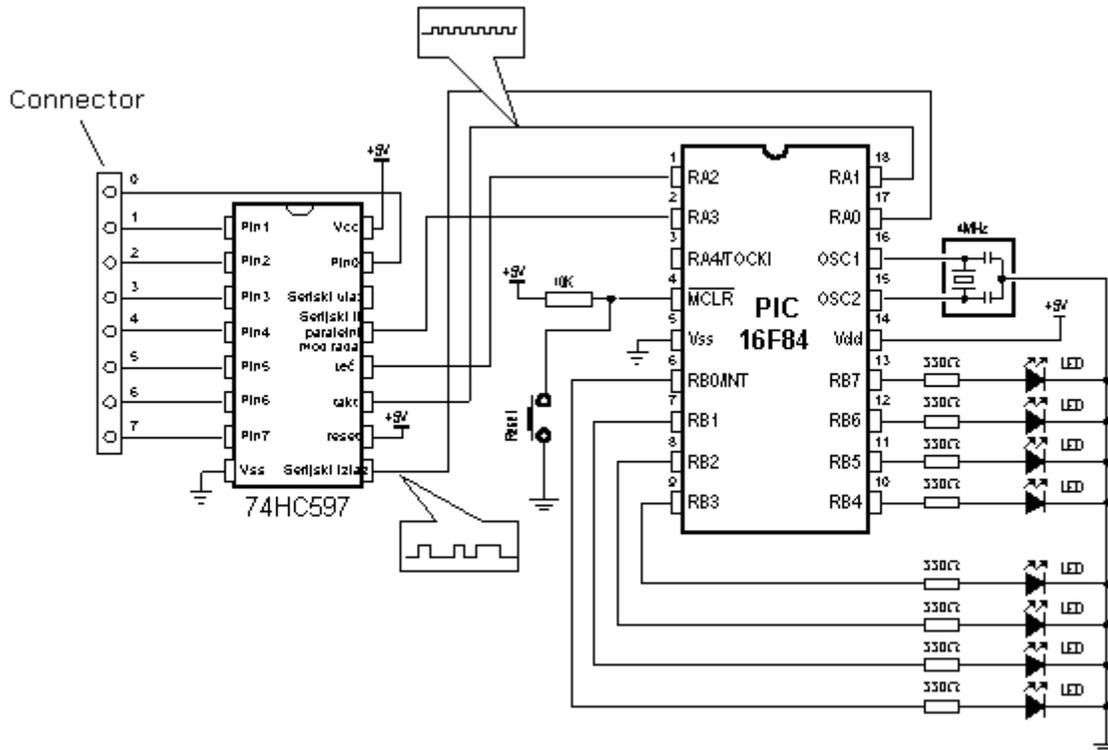
Există două tipuri de regiștrii de deplasare: **de intrare și de ieșire**. **Regiștrii de intrare** încarcă datele paralele, prin intermediul a 8 linii, și apoi le trimite serial prin intermediul a două linii către microcontroler. **Regiștrii de ieșire** operează în direcție opusă: primesc date serial și la un semnal pe linia „latch”, transformă datele în date paralele. Regiștrii de deplasare sunt folosiți în general pentru a mări numărul de intrări – ieșiri ale unui microcontroler. Ei nu prea mai sunt folosiți pentru că microcontrolerele moderne au un număr mare de linii intrare – ieșire. Oricum, utilizarea lor cu microcontrolere cum ar fi PIC16F84 este foarte importantă.

Regiștrii de deplasare de intrare 74HC597

Regiștrii de deplasare de intrare transformă datele paralele în date seriale și le transferă către microcontroler. Modul lor de funcționare este simplu. Sunt patru linii pentru transferul datelor: **clock, latch, load și data**. Datele sunt citite de la pinii de intrare de un registru intern prin intermediul unui semnal „latch”. Apoi, cu un semnal „load”, datele sunt transferate de la registru „latch” de intrare către registru de deplasare, iar de acolo sunt transferate serial către un microcontroler prin intermediul liniilor „data” și „clock”.



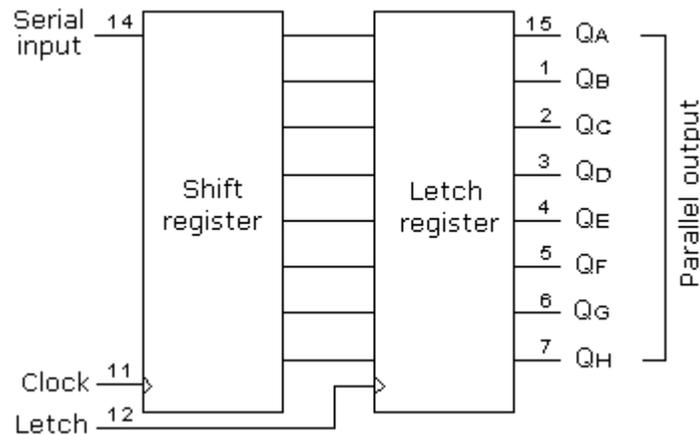
O schemă de legătură a registrului de deplasare 74HC597 la un microcontroler este prezentată mai jos:



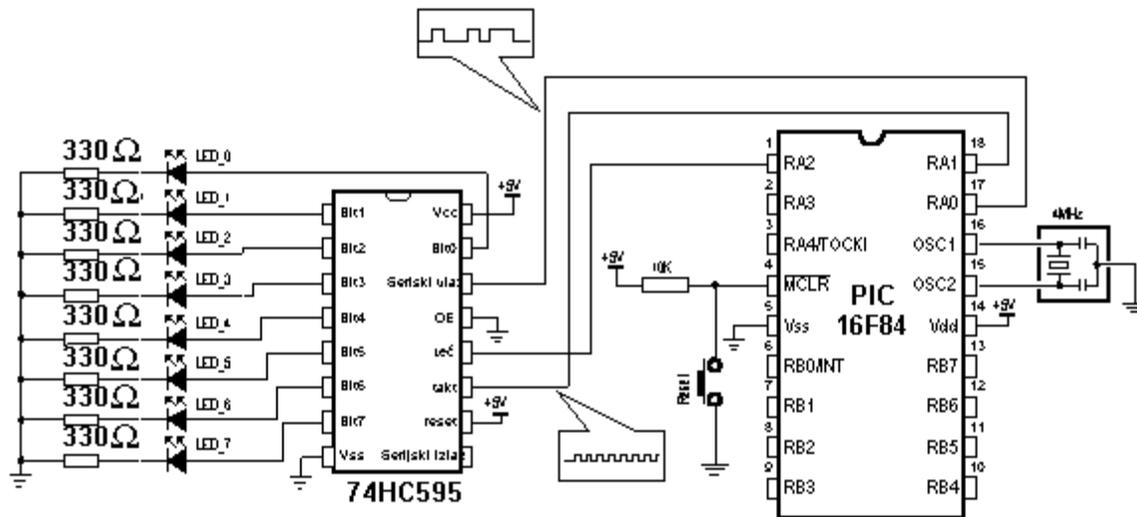
Modalitatea de conectare a unui registru de deplasare de intrare la un microcontroler

Registru de deplasare de ieşire

Regiștrii de deplasare de ieşire transformă datele seriale în date paralele. Pe fiecare front crescător al tactului, registrul de deplasare citește valoarea de la linia de date, o memorează într-un registru temporar, apoi repetă acest ciclu de 8 ori. La un semnal de la linia „latch”, datele sunt copiate din registrul de deplasare în registrul de intrare, apoi datele sunt transformate din date seriale în date paralele.



O schemă a registrului de deplasare este prezentată mai jos:



Conectarea unui registru de deplasare de ieşire la un microcontroler

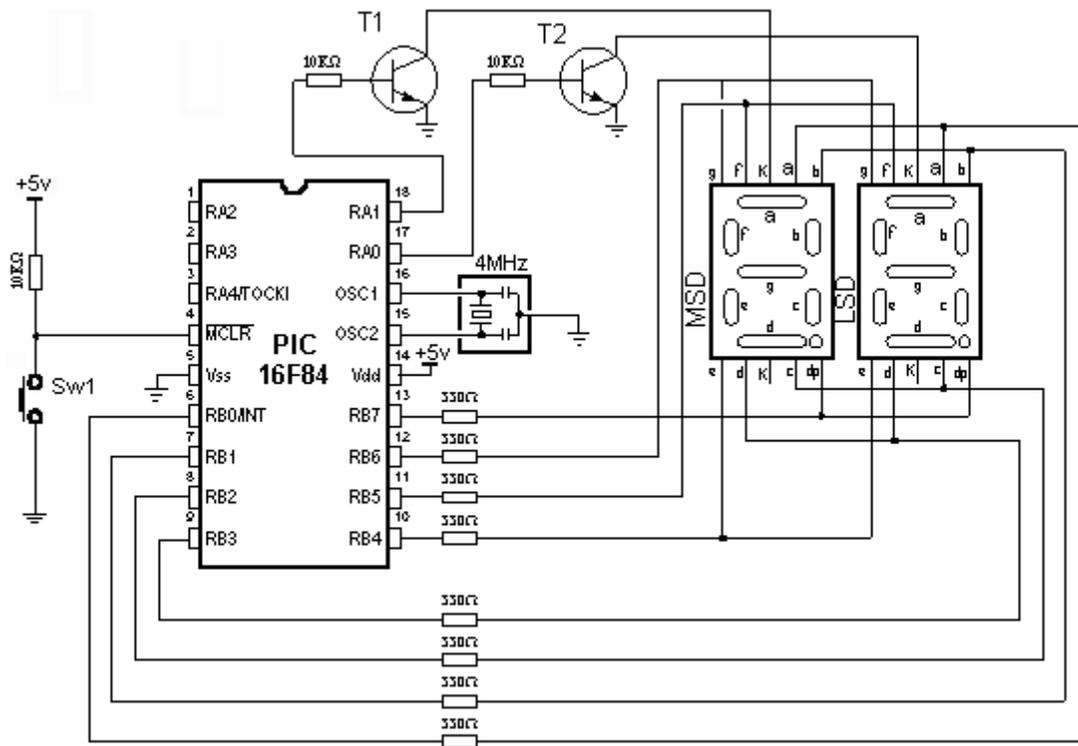
Afișaj cu 7 segmente (multiplexare)

Segmentele într-un afișaj cu 7 segmente sunt aranjate astfel încât să formeze un singur digit de la 0 la F, după cum se observă în desen:



Putem afișa un număr pe mai mulți digiți prin conectarea de afișaje adiționale. Chiar dacă este mult mai confortabil să lucrăm cu LCDuri, afișajele cu 7 segmente sunt încă un standard în industrie. Aceasta din cauza rezistenței la temperatură, vizibilității și unghiului larg de observare. Segmentele sunt marcate cu litere mici: a, b, c, d, e, f, g și dp, unde dp este punctul zecimal. Cele 8 LEDuri din cadrul fiecărui afișaj pot fi aranjate cu catod comun sau cu anod comun. La un afișaj cu catod comun, catodul comun trebuie să fie conectat la linia de 0v și LEDurile sunt activate cu unu logic. Afișajele cu anod comun trebuie să prezinte anodul comun conectat la linia de +5v. Segmentele sunt activate cu zero logic. Dimensiunea afișajului este măsurată în milimetri; se măsoară doar înălțimea digitului (nu carcasa, doar digitul!). Afișajele sunt disponibile cu digiți de înălțimi de 7, 10, 13.5, 20 sau 25 milimetri. Sunt de diferite culori incluzând: roșu, portocaliu și verde. Cea mai simplă metodă pentru a comanda un afișaj este prin intermediul unui driver de afișaj. Acestea sunt disponibile pentru până la 4 afișaje. Alternativ, afișajele pot fi comandate de un microcontroler, și, dacă este necesar mai mult decât un afișaj, metoda de comandare se numește „multiplexare”. Principala diferență dintre cele două metode este numărul de linii de comandă. Un driver special poate avea numai o singură linie de tact și integratul de comandă va accesa toate segmentele și va incrementa afișajul. Dacă avem doar un singur afișaj de comandat de către microcontroler, vor fi necesare 7 linii plus una pentru punctul zecimal. Pentru fiecare afișaj zecimal, este necesară doar câte o linie în plus. Pentru a produce un afișaj cu 4, 5 sau 6 digiți, toate afișajele cu 7 segmente vor fi conectate în paralel. Linia comună (linia catodului comun) este conectată separat și această linie este conectată la zero logic pentru o perioadă scurtă de timp pentru a activa afișajul. Fiecare afișaj este activat de 100 ori pe secundă și vor da impresia că toate afișajele sunt active în același timp. În timp ce fiecare afișaj este activat, informația trebuie livrată astfel încât el va afișa informația corectă. Pot fi accesate până la 6 afișaje în acest mod fără ca strălucirea fiecărui afișaj să fie afectată. Fiecare afișaj este activat efectiv pentru 1/6 din timp și persistența vizuală a ochilor dă impresia că afișajul este pornit tot timpul. Toate semnalele de sincronizare pentru afișaj sunt produse de program, avantajul unui afișaj controlat de un microcontroler este flexibilitatea. Afișajul poate fi configurat ca un contor crescător, contor descrescător, și poate produce un număr de mesaje folosind literele alfabetului care pot fi ușor de afișat.

Exemplul de mai jos arată cum să controlăm două afișaje.



Conectarea unui microcontroler cu afișaje cu 7 segmente în mod multiplexat

Fișierul LED.INC conține două macrouri: LED_Init și LED_Dis2. Primul macro este folosit pentru initializarea afișajului. Aici este definită perioada de reîmprospătare cât și pinii microcontrolerului utilizați pentru conectarea afișajelor.

Exemplu: LED_Dis2 0x34

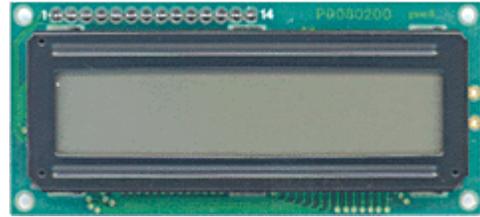
Numărul 34 va fi afișat.

Realizarea macroului arată modalitatea de utilizare a macrourilor într-un program. Programul afișează numărul ,21' în 2 digiți cu **7 segmente**.



Afișaj LCD

Multe dispozitive cu microcontroler folosesc LCDuri inteligente pentru a afișa informația vizuală. Următorul material se ocupă de conectarea unui afișaj LDC Hitachi la un microcontroler PIC. Afișajele LCD proiectate cu HD44780, modulul pentru LCD fabricat de Hitachi, nu sunt scumpe și sunt ușor de folosit, și chiar posibil să producă verificarea datelor afișate folosind cei 8x80 pixeli ai afișajului. Afișajele LCD Hitachi conțin un set de caractere ASCII plus simboluri japoneze, grecești și matematice.



A 16x2 line Hitachi HD44780 display

Fiecare dintre cei 640 de pixeli ai afișajului trebuie să poată fi accesat individual și aceasta se poate realiza cu un număr de integrate SMD pentru control montate pe spatele afișajului. Aceasta ne salvează de o cantitate enormă de fire și de un control adecvat astfel încât sunt necesare doar câteva linii pentru a accesa afișajul. Putem comunica cu afișajul prin intermediul unui bus de date pe 8 biți sau de 4 biți. Pentru un bus de 8 biți, afișajul are nevoie de o tensiune de alimentare de +5v și 11 linii I/O. Pentru un bus de 4 biți sunt necesare doar liniile de alimentare și 7 linii. Când afișajul LCD nu este pornit liniile de date sunt TRI-STATE, ceea ce înseamnă că ele sunt în stare de înaltă impedanță (ca și cum ar fi deconectate) și astfel nu interferează cu funcționabilitatea microcontrolerului când afișajul nu este adresat. LCDul necesită de altfel 3 linii de control de la microcontroler.

Linia **Enable (E)** permite accesul la afișaj prin intermediul liniilor R/W și RS. Când această linie este LOW, LCDul este dezactivat și ignoră semnalele de la R/W și RS. Când linia (E) este HIGH, LCDul verifică starea celor două linii de control și răspunde corespunzător.

Linia **Read/Write (R/W)** stabilește direcția datelor dintre LCD și microcontroler. Când linia este LOW, datele sunt scrise în LCD. Când este HIGH, datele sunt citite de la LCD.

Cu ajutorul liniei **Register select (RS)**, LCD interpretează tipul datelor de pe liniile de date. Când este LOW, o instrucțiune este scrisă în LCD. Când este HIGH, un caracter este scris în LCD.

Citirea datelor de la LCD se realizează similar, cu deosebirea că linia de control R/W trebuie să fie HIGH. Când trimitem un HIGH către LCD, el se va reseta și va accepta instrucțiuni. Instrucțiunile tipice care sunt transmise către un afișaj LCD după reset sunt: pornirea afișajului, activarea cursorului și scrierea caracterelor de la stânga spre dreapta. În momentul în care un LCD este inițializat, el este pregătit să primească date sau instrucțiuni. Dacă recepționează un caracter, el îl va afișa și va muta cursorul un spațiu la dreapta. Cursorul marchează locația următoare unde un caracter va fi afișat. Când dorim să scriem un șir de caractere, mai întâi trebuie să setăm adresa de start, și apoi să trimitem câte un caracter pe rând. Caracterele care pot fi afișate pe ecran sunt memorate în memoria video DD RAM (Data Display RAM). Capacitatea memoriei DD RAM este de 80 bytes.

Afişajul LCD mai conţine 64 bytes CG RAM (Character Generator RAM). Această memorie este rezervată pentru caracterele definite de utilizator. Datele din CG RAM sunt reprezentate sub formă de caractere bitmap de 8 biti. Fiecare caracter ocupă maxim 8 bytes în CG RAM, astfel numărul total de caractere pe care un utilizator poate să le definească este 8. Pentru a afişa caracterul bitmap pe LCD, trebuie setată adresa CG RAM la punctul de start (de obicei 0) şi apoi să fie scrise datele în afişaj. Definirea unui caracter ,special' este exemplificată în figură.

CG RAM address	Bit map	Data
0000		01010
0001		00100
0010		01110
0011		10001
0100		10000
0101		10001
0110		01110
0111		00000

Înainte de a accesa DD RAM, după definirea unui caracter special, programul trebuie să seteze adresa în DD RAM. Orice scriere şi citire a datelor din memoria LCD este realizată de la ultima adresă care a fost setată, folosind instrucţiunea set-address. Odată ce adresa DD RAM este setată, un caracter nou va fi afişat în locul potrivit pe ecran. Până acum am discutat operaţia de scriere şi citire a memoriei unui LCD ca şi cum ar fi o memorie obişnuită. Acest lucru nu este adevărat. Controlerul LCD are nevoie de 40 până la 120 microsecunde (us) pentru scriere şi citire. Alte operaţii pot dura până la 5 ms. În acest timp microcontrolerul nu poate accesa LCDul, astfel un program trebuie să ştie când un LCD este ocupat. Putem rezolva aceasta în două metode.

Set DD RAM address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	A	A	A	A	A	A	A

Set CG RAM address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	A	A	A	A	A	A

Write in data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D	D	D	D	D	D	D	D

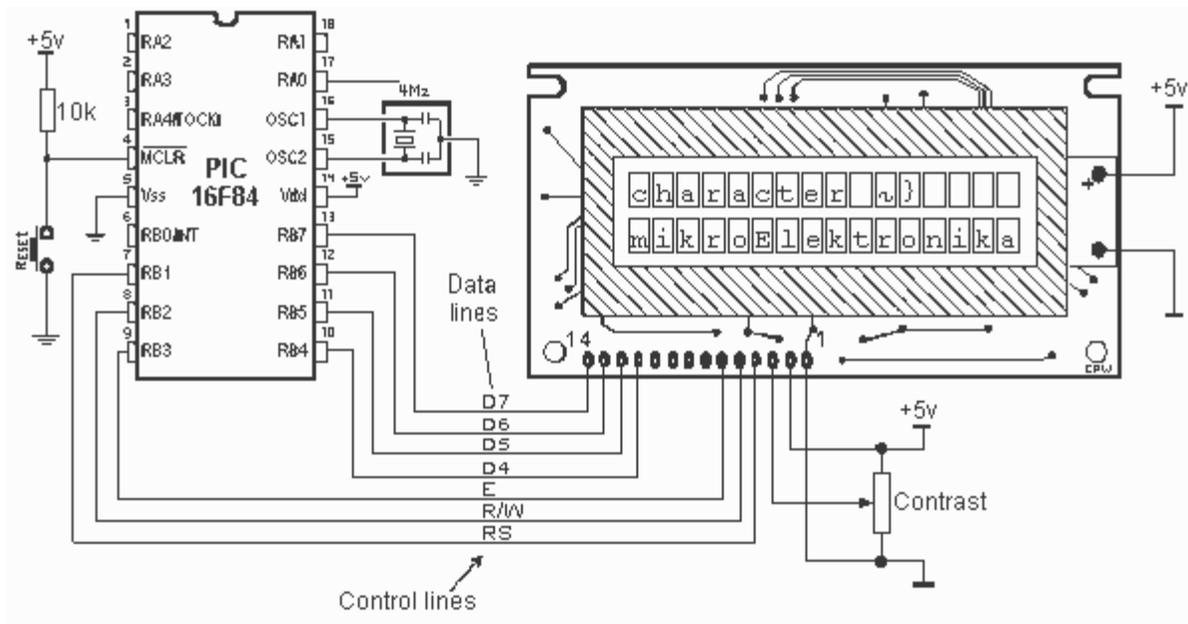
Read data from RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D	D	D	D	D	D	D	D

A=address

D=data

O metodă este verificarea bitului BUSY de pe linia de date D7. Aceasta nu este cea mai bună metodă pentru că LCDul se poate bloca şi programul va sta într-o buclă infinită verificând bitul BUSY. O altă metodă este introducerea unei întârzieri în program. Întârzierea trebuie să fie destul de lungă pentru ca LCDul să termine operaţia în desfăşurare. Instrucţiunile pentru scriere şi citire cu memoria LCDului sunt afişate mai sus. La început am menţionat că avem nevoie de 11 linii I/O pentru a comunica cu un LCD. Oricum, putem comunica cu un LCD printr-un bus de 4 linii. Putem reduce numărul total de linii de comunicaţie la 7. Schema pentru conectarea printr-un bus de 4 biti este în imaginea de mai jos. În acest exemplu folosim un afişaj LCD cu 2x16 caractere, denumit LM16x212 fabricat de producătorul japonez Sharp. Mesajul ,character' este scris pe prima linie urmat de două caractere speciale ,~' şi ,}' . Pe a doua linie este scris cuvântul ,mikroElektronika'.



Conectarea unui afișaj la un microcontroler

Fișierul **LCD.inc** conține un grup de macroui pentru lucrul cu afișajele LCD.

Exemplu: LCDline 2

Când lucrăm cu microcontrolere numerele sunt reprezentate în formă binară. Din această cauză ele nu pot fi afișate. Pentru aceasta este necesar să schimbăm numerele dintr-un sistem binar într-un sistem zecimal pentru ca ele să fie ușor de înțeles. Sursele celor două macroui **LCDval_08** și **LCDval_16** sunt prezentate mai jos.

Macroul **LCDval_08** realizează conversia unui număr binar de 8 biți într-un număr zecimal de la 0 la 255 și îl afișează. Este necesar să declarăm următoarele variabile în programul principal: TEMP1, TEMP2, LO, LO_TEMP, Bcheck. Numărul binar de 8 biți este în variabila LO. Când macroul este executat, echivalentul zecimal al acestui număr este afișat. Zerourile precedente numărului nu sunt afișate.

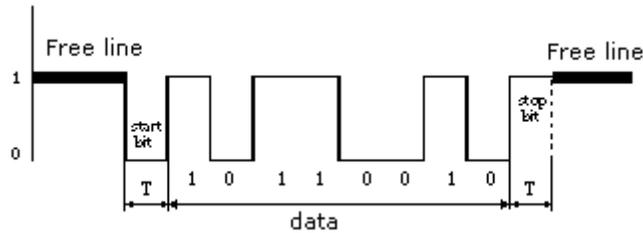
Macroul **LCDval_16** realizează conversia unui număr binar de 16 biți într-un număr zecimal de la 0 la 65535 și îl afișează. Următoarele variabile trebuie declarate în programul principal: TEMP1, TEMP2, TEMP3, LO, HI, LO_TEMP, HI_TEMP, Bcheck. Numărul binar de 16 biți este în variabilele LO și HI. Când macroul este executat, echivalentul zecimal al acestui număr este afișat. Zerourile precedente numărului nu sunt afișate.

Convertor analog – digital de 12 biți

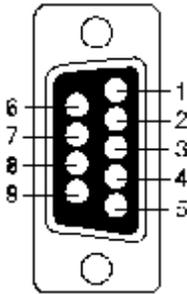
Deoarece totul în lumea microcontrolerelor este reprezentat cu 0 și 1, ce facem cu un semnal care este 0.5 sau 0.77? Aproape toată lumea exterioară unui calculator consistă din semnale analogice. În afară de vorbire și muzică, sunt multe cantități care trebuie cuantificate într-un calculator. Umiditatea, temperatura, presiunea aerului, culoarea, **turbidity**, și concentrația metanului sunt doar o parte. Răspunsul este să luăm un număr de linii digitale pe care să le combinăm astfel încât ele să exprime o valoare analogică. O valoare analogică este orice valoare dintre 0 și 1. O puteți numi „valoare fracțională”. Toate cantitățile de mai sus trebuie să fie convertite la o valoare cuprinsă între 0 și 1 astfel să fie utilizată într-un calculator. Acesta este conceptul cel mai răspândit. El devine ceva mai puțin complex în aplicații. Dacă luăm 8 linii și le aranjăm astfel încât să accepte valori binare, rezultatul total va fi 256 (acesta este obținut de o numărare până la 255 plus valoarea 0). Dacă conectăm aceste 8 linii într-o „cutie neagră”, vor fi numite linii de ieșire și astfel trebuie să-i furnizăm o singură linie de intrare. Cu acest aranjament putem detecta până la 255 incrementări între „0” și „1”. Această cutie neagră este denumită CONVERTOR și pentru că noi convertim din **Analog** în **Digital**, convertorul se numește convertor **AD** sau **ADC (Analog to Digital Converter)**. Convertoarele AD pot fi clasificate după parametri diferiți. Cei mai importanți parametri sunt precizia și modul de transfer al datelor. Referitor la precizie, domeniul este: 8 biți, 10 biți, 12 biți, 14 biți, 16 biți. Deoarece conversia pe 12 biți este un standard industrial, exemplul de mai jos a fost realizat cu ajutorul unui convertor pe 12 biți. Celălalt parametru important este modul prin care datele sunt transferate în microcontroler. Poate fi paralel sau serial. Transmisia paralelă este mai rapidă. Oricum, aceste convertoare sunt de obicei mai scumpe. Transmisia serială este mai lentă, dar considerând prețul scăzut și numărul redus de linii de intrare la un microcontroler, este transmisia preferată pentru multe aplicații. Semnalele analogice pot depăși limitele de intrare ale unui ADC. Acest lucru poate duce la stricarea convertorului. Pentru a proteja intrarea sunt conectate două diode, după cum se observă în schemă. Acestea vor proteja de tensiuni de peste 5V și sub 0V. În exemplul nostru am folosit LTC1286, un ADC pe 12 biți (Linear Technology). Convertorul este conectat cu microcontrolerul prin intermediul a trei linii: data, clock și CS (Chip Select). Linia CS este folosită pentru a selecta un dispozitiv de intrare ca să fie posibil să fie conectate și alte dispozitive de intrare (ex. registru de deplasare de intrare, registru de deplasare de ieșire, ADC serial) pe aceleași linii ale microcontrolerului. Circuitul de mai jos arată cum să conectăm un ADC, tensiune de referință și un afișaj LCD la un microcontroler. Afișajul LCD a fost adăugat pentru a vizualiza rezultatele conversiei AD.

Comunicația serială

SCI este o abreviere pentru Serial Communication Interface, și ca un subsistem special există la majoritatea microcontrolerelor. Când nu este disponibil, cum ar fi în cazul lui PIC16F84, poate fi creat în software.



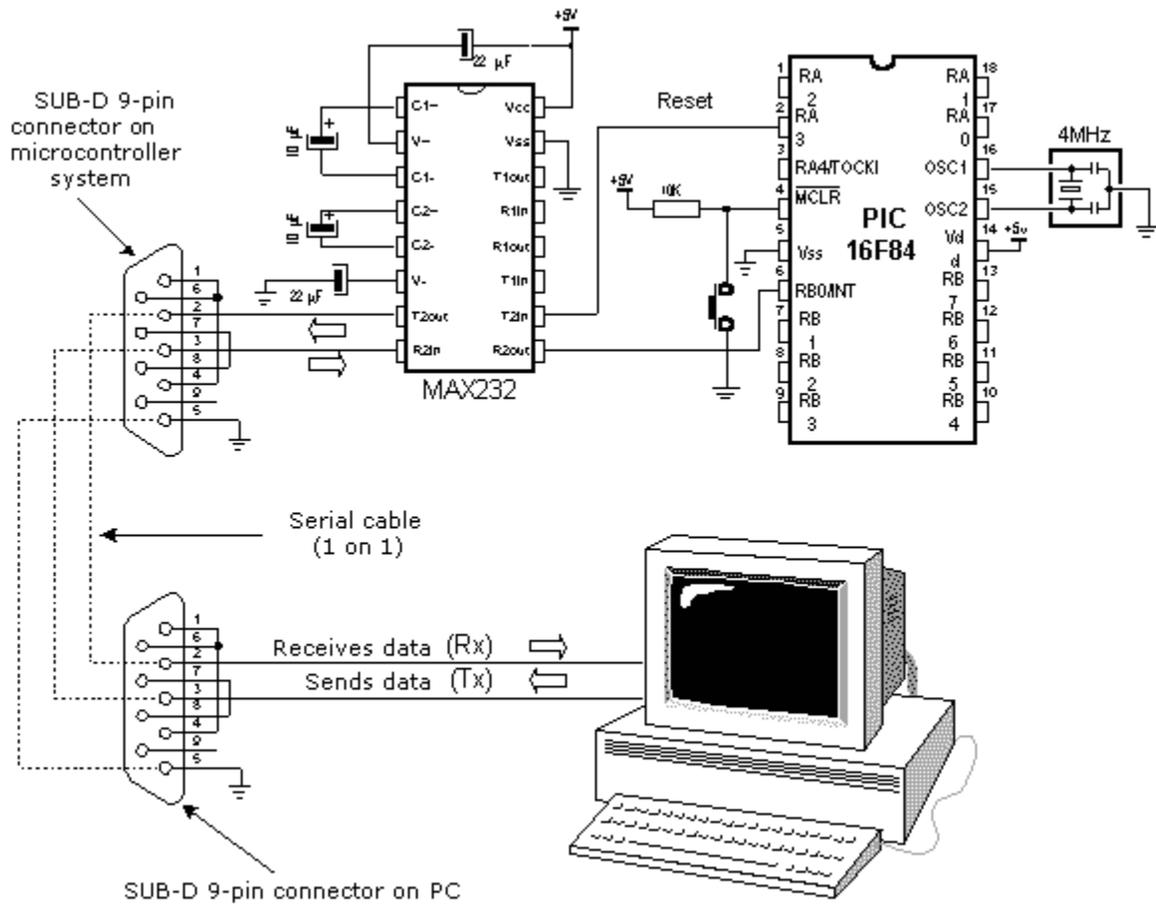
Ca și în cazul comunicației hardware, folosim formatul standard NRZ (Non Return to Zero) cunoscut ca 8 (9)-N-1 sau 8 sau 9 biți de date, fără paritate și cu un bit de stop. Linia liberă este definită starea unu logic. Startul transmisiei – Bitul de Start, are starea zero logic. După biții de date care urmează bitului de start (primul bit este cel mai puțin semnificativ bit) urmează un Bit de Stop care are starea unu logic. Durata bitului de stop ,T' depinde de viteza transmisiei și este ajustat după necesitățile transmisiei. Pentru o viteză de transmisie de 9600 baud, T este 104us.



1. CD (Carrier Detect)
2. RXD (Receive Data)
3. TXD (Transmit Data)
4. DTR (Data terminal Ready)
5. GND (Ground)
6. DSR (Data Set Ready)
7. RTS (Request To Send)
8. CTS (Clear To Send)
9. RI (Ring Indicator)

Descrierea pinilor ai unui conector RS232

Pentru a conecta un microcontroler la un port serial al unui calculator PC, trebuie să ajustăm nivelul semnalelor pentru ca să aibă loc comunicația. Nivelul semnalului la un PC este -10V pentru zero logic și +10V pentru unu logic. Din cauză că nivelul semnalului la un microcontroler este de +5V pentru unu logic și 0V pentru zero logic, avem nevoie de un stadiu intermediar care să realizeze conversia nivelurilor. Un integrat special proiectat pentru această sarcină este MAX232. Schema interfeței este în diagrama de mai jos:



Conectarea unui microcontroler la un PC prin intermediul unei interfețe realizate cu MAX232.

Fișierul RS232.inc conține un grup de macrouri folosite pentru comunicația serială.