

LUCRAREA 9

SISTEM DE AFIȘARE ALFANUMERICĂ PE AFIȘAJ LCD CONTROLAT CU MICROCONTROLLER

1. Introducere

Pentru afișarea caracterelor alfanumerice (consacrate sau definite de utilizator), utilizarea dispozitivelor de afișare cu cristale lichide (LCD) în format matricial este modul cel mai comod și mai răspândit. Fiind o aplicație complexă, managementul datelor afișate trebuie asigurat de un microsistem (cu microprocesor sau microcontroller).

În lucrarea de față se va folosi un afișaj LCD tip L2432 (Seiko), ce permite afișarea caracterelor alfanumerice în format matricial 5x7, pe două linii, cu câte 24 de caractere pe linie. Afișajul permite afișarea a 192 de caractere ASCII, în forma memorată în memoria fixă a generatorului de caractere (CG ROM), precum și a maxim 8 caractere definite de utilizator în forma memorată în memoria volatilă RAM (CG RAM). Cele maxim 8 caractere sunt disponibile la un moment dat al desfășurării programului de afișare. Pe durata întregului program, caracterele pot fi schimbate, modificând conținutul memoriei CG RAM, ceea ce permite utilizarea unui număr mai mare decât 8 caractere (capacitatea de memorare maximă a memoriei RAM a generatorului de caractere definite de utilizator). La elaborarea programului de control al afișajului trebuie ținut cont că simultan pe afișaj nu pot fi văzute decât maxim 8 caractere definite de utilizator.

Pentru controlul afișajului se folosește un sistem de dezvoltare tip PK-HCS12C32 pentru microcontrolere de 16 biți MC9S12C32 (Motorola). Kitul de dezvoltare are și uneltele software necesare dezvoltării programelor, cum ar fi compilatorul pentru limbaj de programare C/C++, linkerul, debuggerul etc. Cu resursele sistemului de dezvoltare, aplicația studiată în lucrarea de laborator va fi dezvoltată în limbaj C, cu adaptările necesare funcționării microcontrollerului.

2.1 Schema bloc a dispozitivului de afișare L2432

În figura 9.1 este dată schema bloc a dispozitivului de afișare L2432 (Seiko).

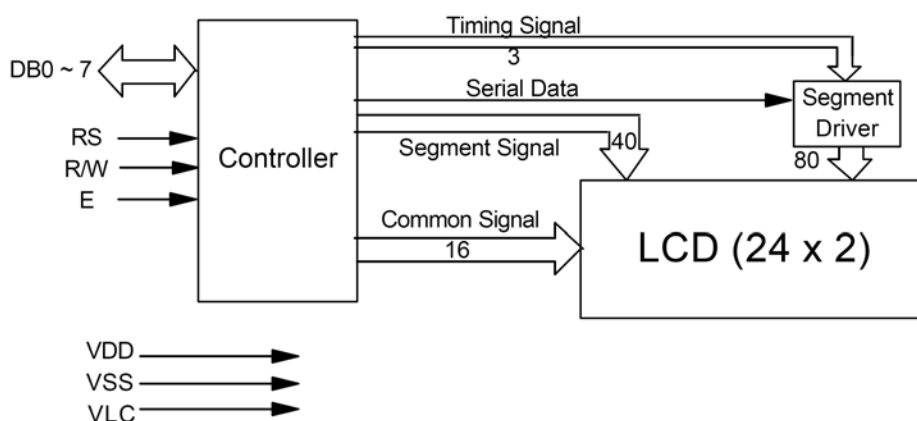


Fig. 9.1. Schema bloc a dispozitivului de afișare L2432 (Seiko)

Dispozitivul de afișare se conectează la microsistemul pe care îl deservește prin intermediul magistralei de date DB0~7 (bilaterală, se pot scrie/citi date în/din controlerul dispozitivului de afișare) și prin intermediul a trei linii de control, astfel:

- RS - selecție registru (*Register Selection*). Semnalul de pe această linie selectează tipul registrului intern din controlerul dispozitivului de afișare (registru de date sau registru de instrucțiuni);

- R/W - citește/scrie (*Read/Write*). Semnalul de pe această linie precizează dacă operațiunea cu registrele interne ale dispozitivului de afișare este scriere sau citire;
- E - autorizare (*Enable*). Semnalul de pe această linie activează sau dezactivează conectarea controlerului dispozitivului de afișare la microsistem (pentru transmiterea de date sau de instrucțiuni).

Pentru alimentarea dispozitivului de afișare sunt prevăzute două linii: VDD (conectată la o tensiune cu valoarea tipică 5 V) și VSS (conectată la masă – GND).

Mai există o linie notată VLC - tensiune de comandă cristale lichide (*Liquid crystal driving voltage*). Tensiunea pe această linie variază între 0 și VDD. Funcție de valoarea acestei tensiuni, se va controla contrastul afișajului. Dacă linia este lăsată în aer (neconectată), pe afișaj nu apare niciun caracter, chiar dacă toate celelalte condiții de programare și semnal sunt corect îndeplinite.

Controlerul dispozitivului de afișare realizează secvențele de semnale necesare controlului panoului cu matricile de afișare de tip 5x7 puncte, conform pozițiilor pe care trebuie scrise informațiile.

2.2 Schema bloc a controlerului KS0066

În figura 9.2 se dă schema bloc a controlerului KS0066 folosit în sistemul de afișare. Semnificația diferitelor blocuri va rezulta din capitolele ce urmează.

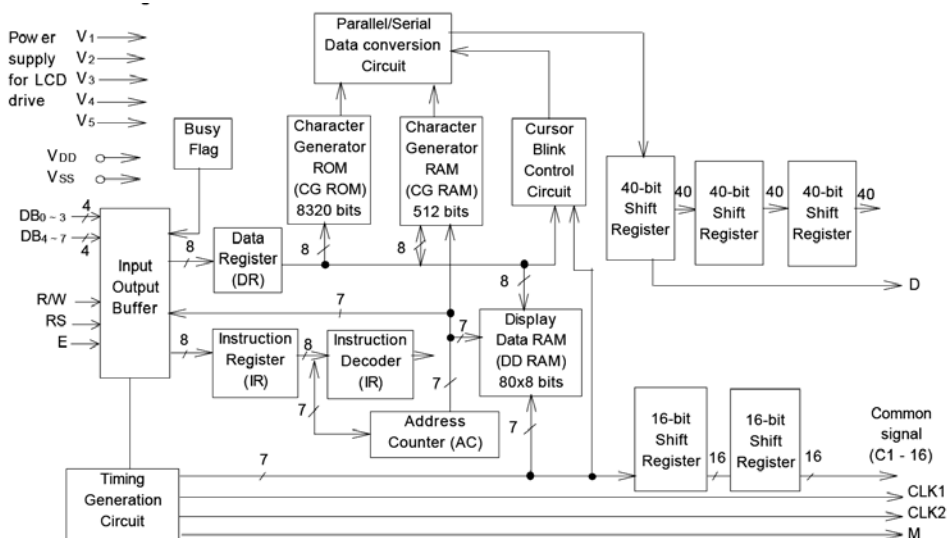


Fig. 9.2. Schema bloc a controlerului KS0066

2.3 Instrucțiuni de utilizare

Registre

Controlerul (KS0066) are două tipuri de registre de 8 biți: *registru de instrucțiuni* (IR) și *registru de date* (DR). Acestea sunt selectate cu semnalul *selecție registru* (RS), ca în Tabelul 1. Registrul de instrucțiuni IR stochează codurile de instrucțiuni, cum ar fi “ștergere afișaj” sau “cursor home” și informațiile de adresă a datelor RAM afișate (DD RAM) sau generatorul de caractere (CG RAM). Registrul de instrucțiuni poate fi scris de către unitatea cu microcontroler, dar nu poate fi citit. Registrul de date DR stochează temporar datele ce trebuie scrise în DD RAM sau CG RAM sau datele citite de la DD RAM sau CG RAM. Pentru a scrie date, datele scrise în registrul de date DR de către unitatea cu microcontroler sunt în mod automat scrise în DD RAM sau CG RAM printr-o operație internă. Pentru a citi date, atunci când adresa datei este scrisă în registrul de instrucțiuni IR, data corespunzătoare este extrasă din registrul de date DR printr-o operație internă. Apoi unitatea cu microcontroler citește data din registrul de date DR. După operația de citire, se fixează următoarea adresă și data din DD RAM sau CG RAM este citită de la adresă și preluată de registrul de date DR pentru următoarea operație de citire.

Tabelul 1. *Selecția registrului*

RS	R/W	Operație
0	0	Selecție IR, scrie IR. Operație internă: șterge afișaj, cursor la capăt
0	1	Citire flag “busy” (ocupat) (DB7) și adresă contor (DB0 la DB7)
1	0	Selecție DR, scrie DR. Operație internă DR în DD RAM sau CG RAM
1	1	Selecție DR, citește DR. Operație internă DD RAM sau CG RAM în DR

Flagul Busy (ocupat) (BF)

Flagul busy indică dacă modulul este gata să accepte următoarea instrucțiune. După cum se arată în Tabelul 1, semnalul este scris la DB7 dacă RS = 0 și R/W = 1. Dacă starea este 1, modulul este în ciclul de funcționare internă și instrucțiunea nu poate fi acceptată. Dacă flagul busy este în starea 0, se poate scrie următoarea instrucțiune. Ca urmare, starea flagului busy trebuie testată înainte de executarea unei instrucțiuni. Pentru a executa o instrucțiune fără a testa starea flagului, trebuie să se aștepte mai mult decât timpul necesar executării instrucțiunii precedente. În capitolul “Descrierea instrucțiunilor” se găsește timpul necesar fiecărei instrucțiuni.

Contorul de adrese (AC) - Address Counter

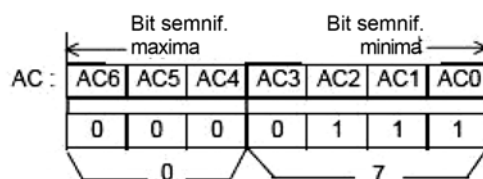
Contorul de adrese (AC) precizează o adresă atunci când data este scrisă în DD RAM sau CG RAM și când este extrasă prin citire data stocată în DD RAM sau CG RAM. Dacă o instrucțiune de setare adresă (pentru DD RAM sau CG RAM) este scrisă în registrul de instrucțiuni IR, informația despre adresă este transferată din registrul de instrucțiuni IR în contorul de adrese AC. Când datele afișate sunt scrise în sau citite din DD RAM sau CG RAM, contorul de adrese AC este incrementat sau decrementat automat, conform cu modul de introducere fixat. Conținutul contorului de adrese AC este extras la ieșirile DB0 la DB6 dacă RS = 0 și R/W = 1, după cum se prezintă în Tabelul 1.

Memoria RAM a datelor afișate (DD RAM) - Display Data Random Access Memory

Memoria DD RAM are capacitatea până la 80 x 8 biți, stocând date pentru 80 de caractere cu coduri de 8 biți. Unele zone de stocare ale memoriei DD RAM, nefolosite la afișare, pot fi utilizate ca locații de memorie de date RAM generale.

O adresă DD RAM care urmează a fi înscrisă în contorul de adrese AC se exprimă în hexazecimal astfel: De exemplu, adresa DD RAM “07”

Correspondența între adresa DD RAM și digiții afișați pe panoul LCD este prezentată mai jos:



Adresele locațiilor de memorie DD RAM

Adresele locațiilor de memorie DD RAM și digiții afișați pe panoul LCD sunt în următoarea corespondență pentru modulele LCD comandate cu factor 1/16. Un controler poate afișa maxim 80 de caractere.

	1	2	3	15	16	17	18	19	20	38	39	40	Digitul afișat
Linia 1	00	01	02	0E	0F	10	11	12	13	25	26	27	DD RAM
Linia 2	40	41	42	4E	4F	50	51	52	53	65	66	67	Adresa (HEX)

Se observă că ultima adresă de pe prima linie (27H) și adresa de început a celei de-a doua (40H) nu sunt consecutive.

Adresele locațiilor de memorie DD RAM pentru modulul de afișaj L2432 folosit în cadrul lucrării de față sunt prezentate mai jos.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Digitul afișat
Linia 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	DD RAM
Linia 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	Adresa (HEX)

Generatorul de caractere ROM (CG ROM) – Character Generator Read Only Memory

Memoria CG ROM generează șabloanele caracterelor pentru matricea de 5x7 puncte pornind de la codurile de 8 biți ale caracterelor. În modulele cu caractere în format matricial 5x7 puncte, generatorul de caractere CG generează 192 de tipuri de caractere.

Tabelul 2 prezintă corespondența între codurile caracterelor din CG ROM și șabloanele caracterelor respective în format matricial de 5x7 puncte.

Generatorul de caractere RAM (CG RAM) – Character Generator Random Access Memory

Memoria CG RAM este folosită pentru a crea prin program șabloane în mod liber. În memoria CG RAM pot fi scrise șabloane în format matricial de 5x7 puncte pentru 8 tipuri de caractere. În Tabelul 3 se prezintă șabloane de caractere create “R” și “¥”), precizându-se adresele CG RAM și datele CG RAM.

Tabelul 2 Corespondența între codurile caracterelor din CG ROM și șabloanele caracterelor în format matricial 5x7 puncte afișate

4 biți superiori 4 biți inferiori	0000	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
XXXX0000	CG RAM (1)														
XXXX0001	(2)														
XXXX0010	(3)														
XXXX0011	(4)														
XXXX0100	(5)														
XXXX0101	(6)														
XXXX0110	(7)														
XXXX0111	(8)														
XXXX1000	(1)														
XXXX1001	(2)														
XXXX1010	(3)														
XXXX1011	(4)														
XXXX1100	(5)														
XXXX1101	(6)														
XXXX1110	(7)														
XXXX1111	(8)														

Pentru a afișa un caracter cu șablon creat, codul caracterului scris în coloana din stânga a tabelului este scris în DD RAM în corespondență cu poziția pe afișaj (digit). Zonele nefolosite pentru afișare sunt disponibile ca locații RAM de memorie de date generale.

Cod caracter (data DD RAM)								Adresa CG RAM						Șablon caracter (data CG RAM)								
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
← bit semnif. maxima				bit semnif. minima →				← bit semnif. maxima			bit semnif. minima →			← bit semnif. maxima				bit semnif. minima →				
0	0	0	0	*	0	0	0	0	0	0	0	0	0	*	*	*	1	1	1	1	0	
											0	0	1	↑				1	0	0	0	1
											0	1	0					1	0	0	0	1
											0	1	1					1	1	1	1	0
											1	0	0					1	0	1	0	0
											1	0	1					1	0	0	1	0
											1	1	0					1	0	0	0	1
											1	1	1		*	*	*	0	0	0	0	0
0	0	0	0	*	0	0	1	0	0	1	0	0	0	↑	*	*	*	1	0	0	0	1
											0	0	1					0	0	1	0	
											0	1	0					1	1	1	1	
											0	1	1					0	0	1	0	0
											1	0	0					1	1	1	1	
											1	0	1					0	0	1	0	0
											1	1	0					0	0	1	0	0
											1	1	1	*	*	*	0	0	0	0	0	
0	0	0	0	*	1	1	1	0	0	0	0	0	0	↑	*	*	*					
											0	0	1									
											0	1	0									
											1	0	1									
											1	1	0									
											1	1	1									

* - valoare indiferentă a bitului

Note:

- În datele CG RAM, "1" înseamnă selecția afișajului, iar "0" neselectia;
- Biții codurilor de caracter 0 la 2 și biții de adresă CG RAM 3 la 5 sunt în corespondență (3 biți - 8 octeți);
- Biții de adresă CG RAM 0 la 2 specifică poziția liniei pentru șablonul caracterului, linia 8 a șablonului este poziția cursorului, pentru care se afișează rezultatul funcției logice SAU între cursor și data CG RAM. Pentru a afișa cursorul, se fixează la "0" data liniei 8. Dacă data este schimbată în "1", bitul corespunzător este afișat, indiferent de cursor;
- Pozițiile coloanelor șablonului caracterului corespund biților de date 0 la 4 din CG RAM, iar bitul 4 se plasează la capătul din stânga. Biți 5 la 7 din CG RAM nu sunt afișați, dar pot fi folosiți ca date RAM de tip general.
- Când se lucrează cu șablonul unui caracter din CG RAM, se fixează în "0" toți biții 4 la 7 din codul caracterului. Biții 0 la 2 determină care șablon va fi extras. Deoarece bitul 3 nu este valabil, 00H și 08H selectează același caracter.

Circuitul de control al cursorului/licărire - *Cursor/Blink Control Circuit*

Circuitul poate genera cursorul sau efectul de licărire. Când numărătorul de adrese (AC) selectează adresa DD RAM, pe poziția digitului corespunzător adresei apare cursorul sau efect de licărire. Când contorul de adresă AC are conținutul 08H, cursorul sau efectul de licărire apare pe poziția digitului 9 de pe linia 1, cum se prezintă mai jos. Cursorul sau efectul de licărire apar și atunci când se selectează memoria RAM a generatorului de caractere (CG RAM) de către numărătorul de adrese. În acest caz cursorul sau efectul de licărire nu au nicio semnificație.

* : bit indiferent

I/D=1: incrementare B=1: licărire ON N=1: umplere 1/16
 I/D=0: decrementare B=0: licărire OFF N=0: umplere 1/8 sau 1/11

A_{CG} : adresă CG RAM

A_{DD} : adresă DD RAM

A_C : adresă GC RAM

S=1: deplasare afișaj S/C=1 deplasare afișaj F=1: matrice 5x10 puncte
 S=0: fără deplasare afișaj S/C=1 mișcare cursor F=0: matrice 5x7 puncte

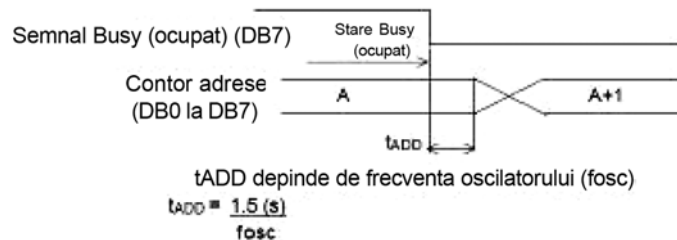
D=1: afișaj ON R/L=1: deplasare dreapta BF=1: operație internă în desfăș.
 D=0: afișaj OFF R/L=0: deplasare stânga BF=0: se poate accepta instrucțiune
 C=1: cursor ON DL=1: 8 biți
 C=0: cursor OFF DL=0: 4 biți

** Timpul de execuție din tabelul de mai sus indică valoarea maximă când frecvența oscilatorului f_{osc} este 250 kHz. Când f_{osc} se schimbă, se schimbă și timpul. De exemplu, dacă $f_{osc} = 270$ kHz, timpul de execuție este $40 \mu s \times 250/270 = 37 \mu s$.

Când controlerul execută o instrucțiune și este în desfășurare o operațiune internă, controlerul nu va accepta și executa nicio altă instrucțiune, cu excepția instrucțiunii Flag Ocupat / Citire adresă (*Busy Flag / Address Read*).

Deoarece flagul “ocupat” (*busy*) este setat la valoarea “1” când o instrucțiune este în curs de execuție, înainte de a trimite de la microprocesor spre controler o altă instrucțiune se testează starea flagului și trebuie să ne asigurăm ca aceasta este “0”.

Pentru a transmite instrucțiuni fără testarea flagului “ocupat” (*busy*) trebuie să ne asigurăm că intervalul de timp dintre două instrucțiuni este mai mare decât timpul de execuție al instrucțiunii precedente.

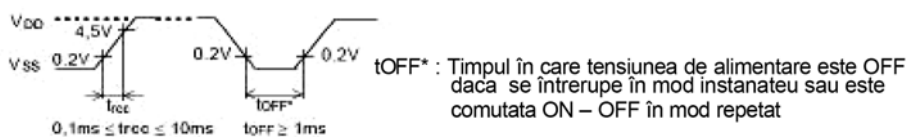


După executarea unei operațiuni de scriere/citire în/din CG/DD RAM, contorul de adrese al memoriei RAM este incrementat sau decrementat cu o unitate în mod automat. Timpul de la frontul căzător al flagului “ocupat” (*busy*) până la sfârșitul reînnoirii contorului de adrese (t_{ADD}) este prezentat mai jos.

4. Inițializare

4.1 Inițializare automată

Sistemul este inițializat automat la stabilirea alimentării, dacă sunt îndeplinite următoarele condiții pentru sursa de alimentare:



Într-o inițializare automată se execută următoarele instrucțiuni:

- ștergere afișaj

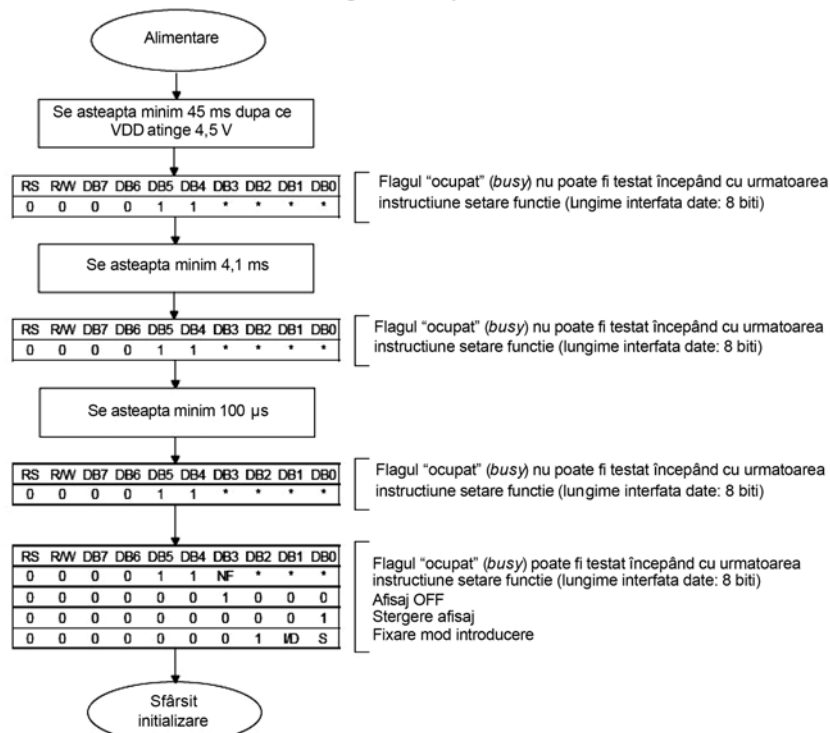
- setare funcție
DL = 1 : lungimea interfeței de date: 8 biți
N = 0, F = 0 : umplere 1/8, font caracter: matrice de 5x7 puncte
- control afișaj ON/OFF
D = 0 : afișaj OFF
C = 0 : cursor OFF
B = 0 : licărire OFF
- fixare mod introducere
I/D = 1 : incrementare
S = 0 : afișajul nu se deplasează

Deoarece unele condiții fixate la inițializare pot fi neconvenabile în aplicația modulului afișaj LCD, ulterior se execută instrucțiunea de setare funcție. Flagul "busy" indică ocupat până se termină inițializarea. Starea "ocupat" se menține încă 20 ms după ce VDD atinge 4,5 V. Când condițiile pentru sursa de alimentare nu sunt îndeplinite, inițializarea automată nu se execută. Inițializarea se execută în acest caz folosind instrucțiuni, ca în paragraful 4.2 Inițializare cu instrucțiuni.

4.2 Inițializare cu instrucțiuni

Dacă inițializarea automată nu a fost executată datorită neîndeplinirii condițiilor necesare de către sursa de alimentare, se folosește o interfață de date de 8 sau de 4 biți (funcție de aplicație) și instrucțiunile din Tabelul 4 și Tabelul 5 pentru inițializare. Cum nu se poate ști dacă lungimea interfeței de date a fost setată la 8 sau 4 biți la conectarea alimentării, se execută de două ori instrucțiunea "setare funcție" pentru a fixa lungimea interfeței de date la 8 biți și apoi se fixează lungimea dorită pentru interfața de date, prin executarea adecvată a instrucțiunii "setare funcție".

Tabelul 5 Lungimea interfeței de date – 8 biți



5. Descrierea instrucțiunilor

5.1 Ștergere afișaj

Șterge afișajul și aduce cursorul pe poziția inițială (home) la adresa 0. La toate adresele DDRAM se scrie codul "spațiu" 20h, iar în contorul de adrese (AC) se înscrie adresa DDRAM 0, iar

dacă a fost mutat, displayul revine pe poziția inițială. Cursorul revine pe capătul din stânga al liniei 1, cu excepția M4024.

În cazul M4024, când cursorul sau clipirea sunt pe linia 3 sau 4, acesta revine la capătul din stânga al liniei 3. După executarea instrucțiunii ștergere afișaj, se setează bitul I/D = 1 (incrementare) în funcțiile mod de intrare.

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

5.2 Aducere cursor pe poziția inițială

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

Aduce cursorul pe pozitia inițială (*home*). În contorul de adrese (AC) se setează adresa DDRAM 0. Afișajul deplasat revine pe poziția inițială (*home*) și conținutul DDRAM nu se schimbă. Când cursorul sau clipirea este ON, el revine pe capătul din stânga al liniei 1, cu excepția M4024. În cazul M4024, când cursorul sau clipirea sunt pe linia 3 sau 4, atunci acesta revine la capătul din stânga al liniei 3.

5.3 Setare mod intrare

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

Fixează direcția deplasării cursorului și dacă informația afișată pe display se deplasează atunci când datele sunt scrise sau citite.

I/D: adresa DDRAM este incrementată sau decrementată cu 1 când un cod caracter este scris în sau citit din DDRAM (valabil pentru scriere/citire în/din CGRAM).

Când I/D = 1, adresa este incrementată cu o unitate și cursorul sau clipirea se deplasează spre dreapta.

Când I/D = 0, adresa este decrementată cu o unitate și cursorul sau clipirea se deplasează spre stânga.

S: dacă S = 1, întregul display este shiftat spre dreapta sau stânga pentru scriere în DDRAM. Poziția cursorului nu se schimbă, numai conținutul afișajului se mută. Nu se execută mutare display pentru citire din DDRAM.

Când S = 1 și I/D = 1, displayul se mută cu un digit spre stânga după ce data a fost scrisă în DDRAM.

Când S = 1 și I/D = 0, displayul se mută cu un digit spre dreapta după ce data a fost scrisă în DDRAM.

Dacă S = 0 nu apare shiftare a afișajului.

5.4 Control afișaj ON/OFF

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

Asigură comutarea ON/OFF a întregului afișaj, a cursorului și controlează clipirea cursorului.

D: Când D = 1, displayul este ON

Când D = 0, displayul este OFF.

Dacă se folosește D = 0, datele afișate rămân în DDRAM. Datele pot fi afișate din nou fixând D = 1.

C: Când C = 1, cursorul este afișat;

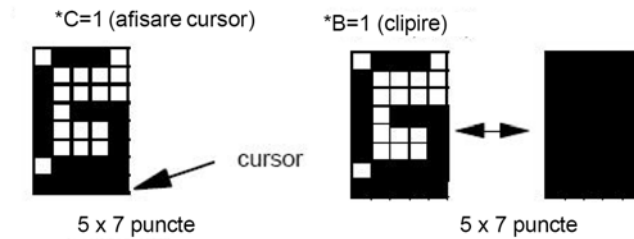
Când C = 0, cursorul nu este afișat;

Cursorul este afișat pe linia de puncte de sub fonturi.

B: Când B = 1, caracterul de pe poziția cursorului începe să clipească;

Când B = 0, caracterul nu clipește.

Pentru a clipi, toate punctele negre și caracterul sunt comutate la fiecare cca 0,4 s, pentru frecvența a oscilatorului = 250 kHz. Cursorul și clipirea pot fi active în același timp.



5.5 Mutare (shiftare) display/cursor

	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Cod	0	0	0	0	0	1	S/C	R/L	*	*	*: valoare indiferenta bit

Asigură deplasarea cursorului și mutarea afișajului fără modificarea conținutului memoriei DDRAM. Poziția cursorului corespunde conținutului contorului de adrese (AC). Instrucțiunea este utilă la corectarea sau refacerea conținutului afișajului, deoarece poziția cursorului sau a displayului se poate modifica fără a scrie sau citi datele afișate. Pentru un display cu două linii, cursorul este deplasat de pe poziția digitului 40 (adresa DDRAM 27) a liniei 1 pe poziția 1 a liniei 2. Informațiile afișate pe liniile 1 și 2 sunt mutate în timp ce comanda "mutare display" deplasează orizontal conținutul fiecărei linii. Totuși conținutul liniei 1 nu este deplasat pe linia 2 și nici conținutul liniei 2 nu este deplasat pe linia 1;

NOTA: M1641 funcționează intern ca un display de 8 caractere x 2 linii, L1614 ca display cu 32 caractere x 2 linii, L2014 ca display cu 40 caractere x 2 linii și M4024 ca două displayuri cu 40 caractere x 2 linii. (paragraful "Locații de adresă" din documentație).

S/C	R/L	Funcționare
0	0	Poziția cursorului este mutată spre stânga (contorul de adrese este decrementat cu 1)
0	1	Poziția cursorului este mutată spre dreapta (contorul de adrese este incrementat cu 1)
1	0	Întregul display se mută spre stânga împreună cu cursorul
1	1	Întregul display se mută spre dreapta împreună cu cursorul

5.6 Setare funcționare

	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Cod	0	0	0	0	1	DL	N	F	*	*	*: valoare indiferenta bit

Funcția "setare funcționare" fixează lungimea de date a interfeței (4 sau 8 biți paralel), numărul de linii ale displayului și tipul de font caracter (5x7 sau 5x10).

DL: lungimea datei în interfață

Când DL = 1, lungimea datei este fixată la opt biți (DB7 la DB0).

Când DL = 0, lungimea datei este fixată la patru biți (DB7 la DB4). Cu o interfață de 4 biți, mai întâi se transferă cei 4 biți superiori, apoi cei 4 biți inferiori.

N: când N = 1, serviciul este 1/16;

când N = 0, serviciul este 1/8 sau 1/11.

F: fonturile pentru caractere

Când F = 1, fontul pentru caractere este fixat la matrice de 5x10 puncte;

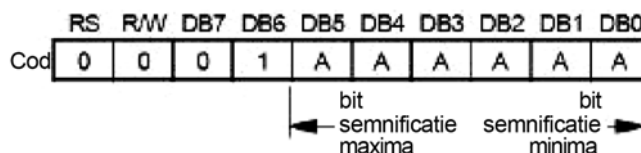
Când F = 0, fontul pentru caractere este fixat la matrice de 5x7 puncte.

Dacă N este fixat la valoarea 1, F devine "indiferent".

N	F	Număr linii afișate	Font caracter	Serviciu	Tip modul LCD
0	0	1	5x7	1/8	-
0	1	1	5x10	1/11	-
1	*	2	5x7	1/16	M1641,M1632,L1642,L1614,L2012,L2022,L2014,L2432,L4042,M4024

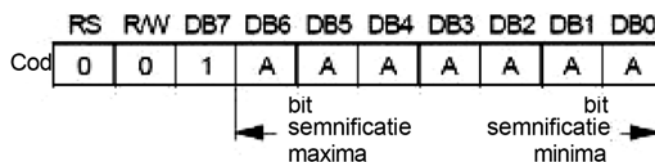
O instrucțiune de setare funcție trebuie să fie executată înainte de toate instrucțiunile, cu excepția instrucțiunii citire flag ocupat / adresă. Dacă se execută mai întâi o altă instrucțiune, nu este efectivă nicio lungime a datelor pentru interfață.

5.7 Fixare adresă CG RAM



Fixarea adresei CGRAM este exprimată binar prin AAAAAA și se transferă contorul de adrese (AC). Data scrisă sau citită de procesor este în sau de la CGRAM. Prima adresă CG RAM este 40h, cand AAAAAA = 000000.

5.9 Fixare adresă DD RAM



Fixarea adresei DDRAM se exprimă binar prin AAAAAAA și se transferă contorul de adrese (AC). Data scrisă sau citită de procesor este în sau de la DD RAM. Când N = 0 (display cu 1 linie), adresele sunt de la 00H la 40H. Când N = 1 (display cu 2 linii: M1632, L1642, L1652, L2012, L2022, L2432, L4042), adresele folosite pentru linia 1 (AAAAAA) sunt de la 00H la 27H, iar cele pentru linia 2 sunt de la 40H la 67H. Pentru M1641, L1614, L2024 și M2024, vezi paragraful "Locații de adresă" din documentație.

5.9 Citire flag ocupat/adresă



Se citește flagul "ocupat", indicând dacă modulul lucrează într-o operațiune intenă din cauza instrucțiunii anterioare.

Când BF = 1, modulul execută o operațiune internă și instrucțiunea următoare nu poate fi acceptată până când BF nu devine 0.

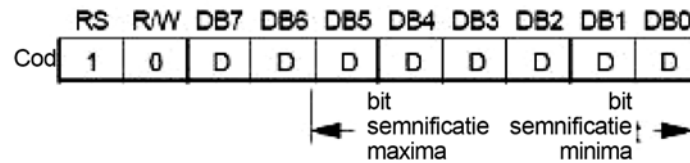
Când BF = 0, instrucțiunea următoare poate fi acceptată. Ca urmare, trebuie ca BF = 0 înainte de a scrie o nouă instrucțiune.

Valoarea binară AAAAAA a contorului de adrese (AC) este citită în același moment ca și flagul "busy". Adresele contorului de adrese sunt folosite atât pentru CG RAM cât și pentru DD RAM, iar instrucțiunea de fixare adresă de dinaintea executării acestei instrucțiuni arată dacă adresa este de la CG RAM sau DD RAM.

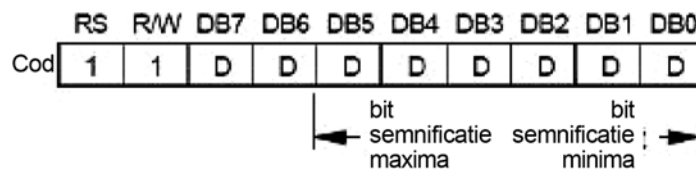
5.10 Scrie date în CG RAM sau DD RAM

Se scrie o dată binară de 8 biți DDDDDDDD în CG RAM sau DD RAM. Adresa CG RAM sau DD RAM fixată înaintea acestei instrucțiuni selectează zona de RAM specifică.

După scriere, adresa este incrementată sau decrementată automat, după cum s-a fixat modul de intrare. Astfel, modul de intrare setat determină dacă displayul se shiftază sau nu după o operațiune de scriere.



5.11 Citește date din CG RAM sau DD RAM



Se citește o dată binară de 8 biți DDDDDDDD din CG RAM sau DD RAM. Adresa CG RAM sau DD RAM fixată înaintea acestei instrucțiuni selectează zona de RAM specifică.

Dacă nu s-a executat nicio instrucțiune de fixare a adresei înaintea unei instrucțiuni de citire, prima dată citită nu este valabilă. Data este în mod normal citită la timpul al doilea dacă se execută citiri consecutive. Pentru DD RAM, dacă s-a executat o instrucțiune de deplasare cursor chiar înainte de citrea DDRAM, nu mai este necesar să se execute o instrucțiune de fixare adresă, deoarece instrucțiunea deplasare cursor realizează acest lucru.

După o operațiune de citire, adresa este în mod automat incrementată sau decrementată, conform cu modul de intrare fixat, dar displayul nu este shiftat conform modului de intrare ales.

NOTA. Contorul de adrese (AC) este automat incrementat sau decrementat cu o unitate conform modului de intrare selectat după ce se execută o instrucțiune de scriere în CGRAM sau DDRAM. Dacă o instrucțiune de citire se execută imediat după o astfel de instrucțiune, informația din RAM specificată de contorul de adrese (AC) nu este extrasă. Data corectă este extrasă în următoarele condiții:

- se execută o instrucțiune de fixare adresă chiar înainte de o instrucțiune de citire;
- pentru DD RAM, o instrucțiune de deplasare a cursorului se execută imediat înainte de o instrucțiune de citire;
- a doua instrucțiune sau următoarele se execută succesiv cu o instrucțiune de citire.

6. Interfațarea cu unități cu microprocesor

Modulele LCD cu controler propriu pot fi interfațate cu unități cu microprocesor de 4 și 8 biți.

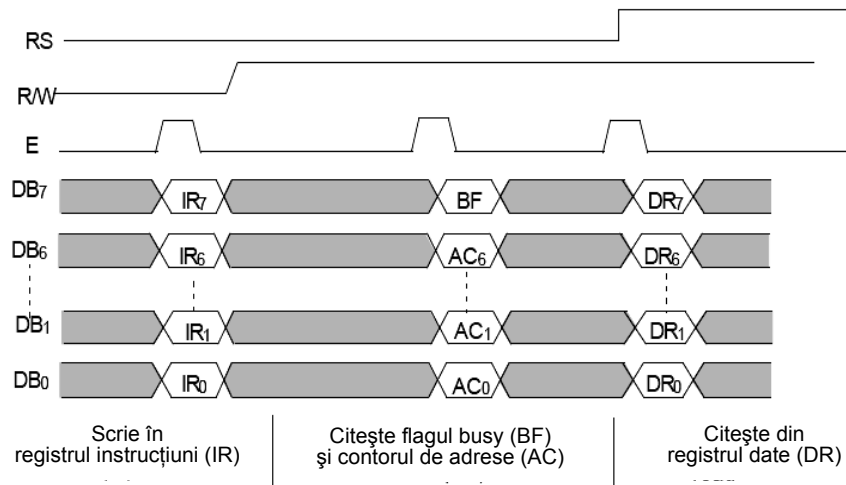
6.1 Interfațare în funcționare pe 8 biți

Când interfața de date este de 8 biți, datele se transferă folosind magistrala de date DB0 la DB7.

7. Sistemul de dezvoltare cu microcontroler

Pentru realizarea aplicației de afișare pe modul matricial LCD s-a folosit kitul SofTec PK-HCS12C32. Sistemul permite dezvoltarea de aplicații pentru microcontrolere de tip Motorola MC9S12C32.

Principalele caracteristici ale microcontrolerului MC9S12C32 sunt:



- nucleu de 16 biți de înaltă performanță;
- memorie FLASH de 32 KB;
- memorie RAM de 2 KB;
- o interfață serială de comunicații asincrone (SCI);
- o interfață serială pentru periferice (SPI)
- un modul timer de 16 biți cu 8 canale (TIM);
- generator de impulsuri cu modulare în durată de 8 biți cu 6 canale (PWM);
- un convertor analog digital de 10 biți cu 8 canale (ADC);
- un modul CAN compatibil software cu standardul CAN 2.0 A, B (MSCAN);
- modul generator reset clock (CRG);
- magistrală cu viteză de 25 MHz;
- domeniul tensiunii de intrare de la 2,97 la 5,5 V;
- mod de depanare cu sistem “single wire” (BDM);
- modul de depanare avansată, incluzând puncte de întrerupere (*breakpoints*) și buffer de modificare a desfășurării programului (DBG12)

Kitul de dezvoltare folosește avantajele mediului de dezvoltare integrat (IDE) Metrowerks CodeWarrior (care grupează Editor, Asamblor, Compilator “C” și Debugger) și ale interfeței Motorola BDM, care permite încărcarea și depanarea aplicației utilizator în memoria FLASH a microcontrolerului.

Împreună cu CodeWarrior, kitul de dezvoltare asigură uneltele necesare pentru a scrie, compila, încărca, emula *in-circuit* și depana codul utilizator. Executarea programului la întreaga viteză permite testarea hardware și software în timp real. Kitul PK-HCS12C32 se conectează la calculatorul gazdă folosind un port USB.

În Figura 9.3 este interfața USB – BDM ce permite încărcarea fișierelor executabile obținute în mediul IDE Metrowerks CodeWarrior (după compilare și linkeditare) și depanarea aplicației (folosind comenzi ale mediului de dezvoltare generate de calculatorul gazdă).

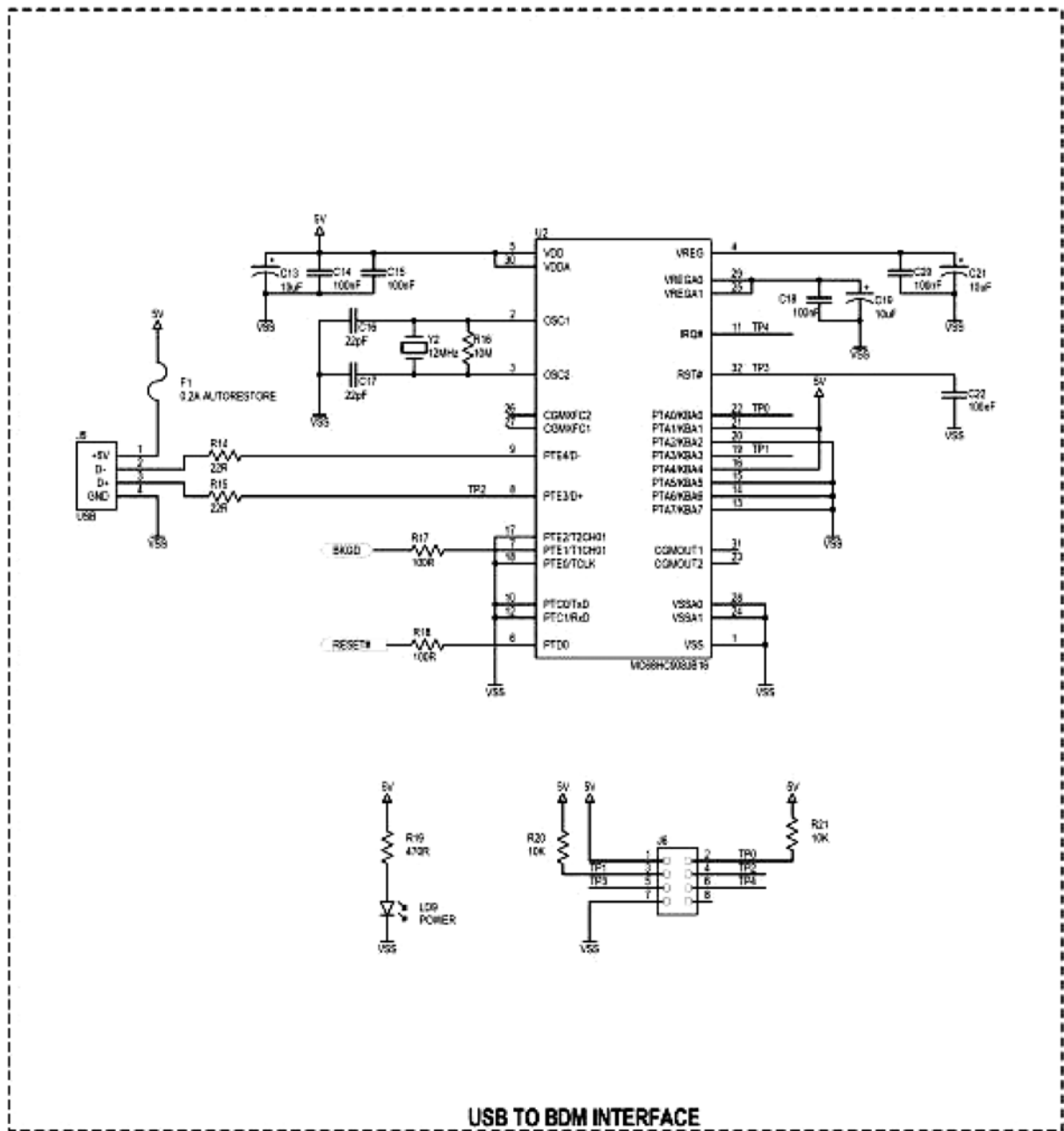


Fig. 9.4 Interfața USB – BDM a sistemului de dezvoltare PK-HCS12C32 cu microcontroler Motorola MC9S12C32

În Figura 9.4 se dă schema electrică a unității centrale cu microcontroler Motorola MC9S12C32, din sistemul de dezvoltare PK-HCS12C32 și modul de conectare a dispozitivului de afișare LCD tip L2432. Datele necesare afișajului LCD sunt vehiculate prin intermediul portului B (PB0, ..., PB7), în timp ce liniile de control al afișajului se conectează în portul A, după cum urmează: linia RS în PA0, linia R/W în PA1 și linia E în PA2.

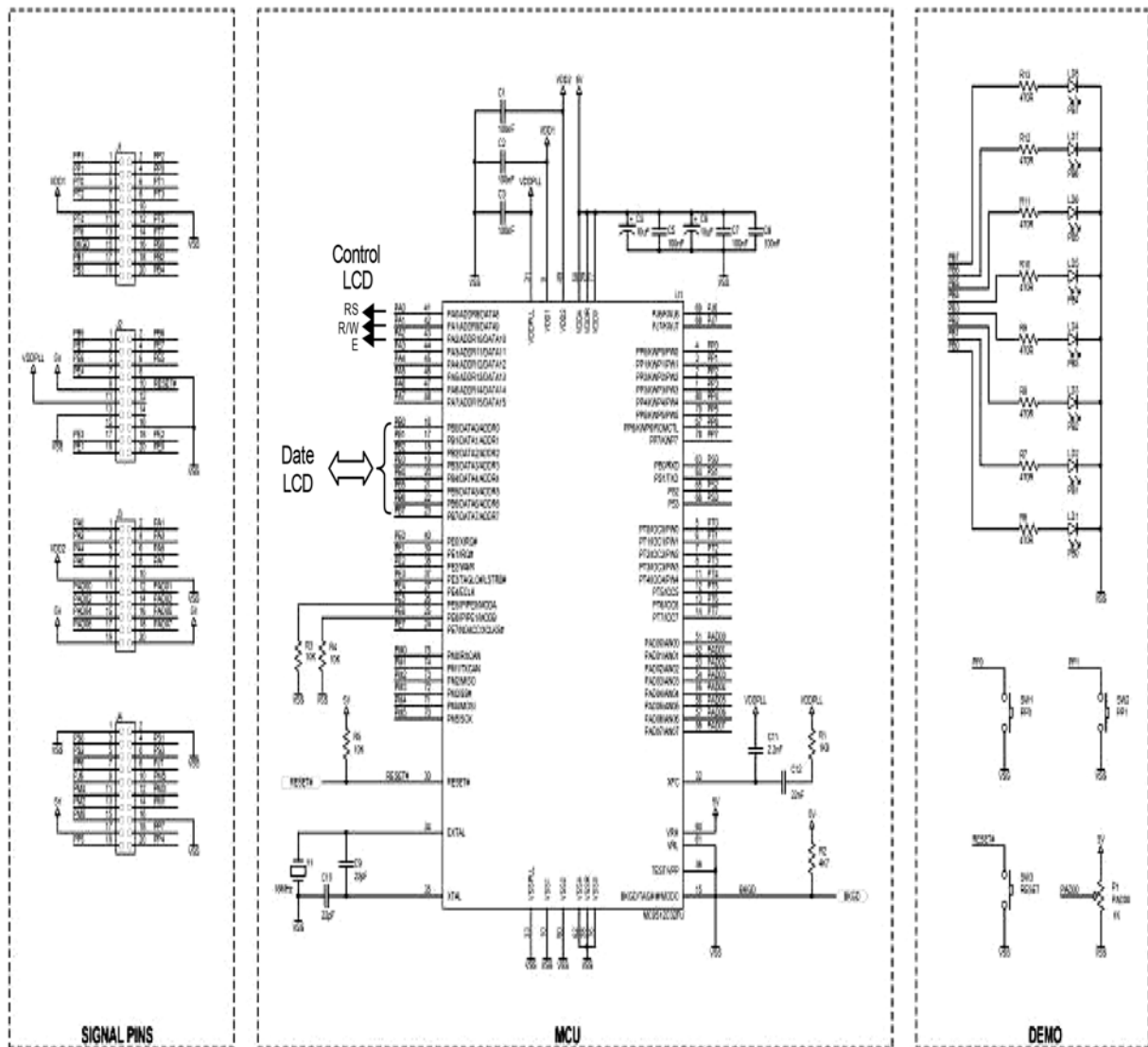


Fig. 9.4 Schema electrică a unității centrale cu microcontroler Motorola MC9S12C32