

2.3. Interfatarea sistemelor EMBEDDED

Un sistem EMBEDDED poate comunica cu unul sau mai multe sisteme, foarte rar prin intermediul unor anumite interfețe de tip paralel interfețe și cel mai frecvent prin interfețe de tip serial, de care ne vom ocupa în detaliu în cele ce urmează.

Transmisia paralelă prezintă avantajul unei rate de transmisie foarte mari dar are și următoarele dezavantaje:

- datorită marginii de zgomot reduse distanța la care se poate asigura calitatea dorită a transmisiei nu poate depăși câțiva metri;
- cost ridicat din cauza numărului mare de fire necesare transmisiei.

2.3.1 Comunicația serie

Generalități

Comunicație de date se referă la transmisia electronică de informații între două sau mai multe sisteme, aflate la distanță unele de altele.

Transmisia serială permite eliminarea neajunsurilor semnalate la transmisia paralelă asigurând transmisii la distanțe mari și foarte mari prin cabluri sau canale radio (prin diverse tehnici de modulație) precum și un cost redus (în cazul legăturilor prin cablu fiind necesare doar 2-3 fire)

În același timp sunt de menționat și problemele pe care le ridică comunicația serie și anume:

- nevoia unei conversii paralel-serie la emisie și a uneia serie-paralel la recepție;
- rezolvarea sincronizării de bit, de simbol și de bloc de date.

Modelul general al unei comunicații serie.

Transmisia serială este de două tipuri: asincronă și sincronă. Comunicația serială asincronă presupune sincronizarea receptorului pentru fiecare caracter transmis prin încadrarea biților de informație ai acestuia cu câte un bit de start și unul de stop. Caracterul asincron nu se referă la modul de lucru al emițătorului și receptorului (sincrone ambele în procesul de transmisie) ci la momentele de apariție a caracterelor ce urmează a fi transmise (câte unul la momente de timp arbitrare, iar între două caractere apar pauze mai mari sau mai mici. Exemplu tipic îl reprezintă comunicația dintre o tastatură și un PC.

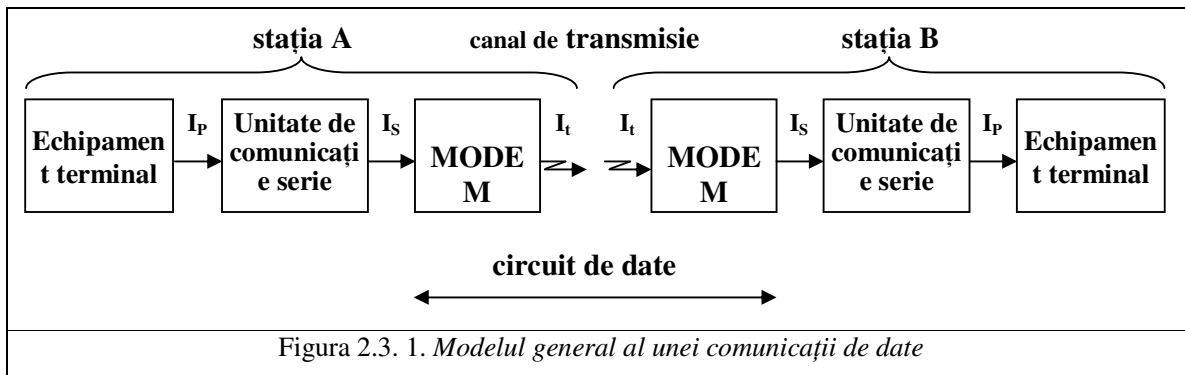
Transmisia sincronă este tipică în cazul transmiterii unor blocuri de caractere (de exemplu o pagină de text) sau a unor blocuri de date reprezentând de exemplu *pixelii* unei imagini. Cerințele față de oscilatorul local al receptorului sunt mai severe ca în cazul comunicației asincrone deoarece în cazul comunicației seriale sincrone oscilatorul local al receptorului trebuie să rămână sincron cu oscilatorul emițătorului pe toată durata recepționării blocului de date.

Schema generală a unei transmisii de date între două stații este prezentată în Figura 2.3. 1.



UNIUNEA EUROPEANĂ

MINISTERUL MUNCII, FAMILIEI ȘI
PROTECȚIEI SOCIALE
AMFOSDRUFONDUL SOCIAL EUROPEAN
POSDRU
2007-2013INSTRUMENTE STRUCTURALE
2007-2013



Cele două stații au structuri similare (pentru exemplificare) și se compun din următoarele elemente:

- Echipamentele terminale (de exemplu procesoare, microcalculatoare sau terminale speciale de date) pot fi surse sau destinație pentru informația transmisă și se interfațează paralel cu unitatea de comunicație serie.
- Unitatea de comunicație serie (controller sau interfață de comunicație) care asigură următoarele funcțiuni: controlul comunicației, conversia paralel serie la emisie respectiv serie paralel la recepție (între interfețele I_s și I_p), detecția erorilor etc.
- *MODEM*-ul cu rolul de a face adaptarea electrică cu linia fizică (canalul) pe care are loc transmisia. În cazul unor canale analogice (de exemplu canale de telefonie sau radio) mai execută modularea și demodularea unui semnal purtător corespunzător lărgimii de bandă a canalului utilizat și a vitezei de transmisie.
- Circuitul de date este porțiunea cuprinsă între cele două unități de comunicație serie și poate asigura comunicația serie a datelor fără a cunoaște detaliile implementării legăturii fizice pe linie (legături între calculatoare la distanțe mici).
- *Moduri de utilizare a legăturii de date.* Există trei moduri de bază de utilizare a unei legături de date și anume
SIMPLEX - transmisie unidirecțională (numai de la A la B sau de la B la A).
SEMDUPLEX - transmisie în ambele sensuri dar succesiv (de exemplu de la A la B iar apoi de la B la A).
DUPLEX - transmisie simultană în ambele sensuri.

2.3.2 Comunicația serială asincronă

Cuvântul de cod corespunzător unui caracter alfanumeric se transmite încadrat de bitul de *START* (zero logic) și de bitul sau biții de *STOP* (având duratele T , $1,5T$ sau $2T$ pe nivelul unui logic). Intervalul între două caractere alfanumerice transmise poate fi arbitrar (pauza), linia de comunicație fiind menținută pe un nivel de tensiune corespunzătoare lui *unu logic*. (vezi Figura 5.10).

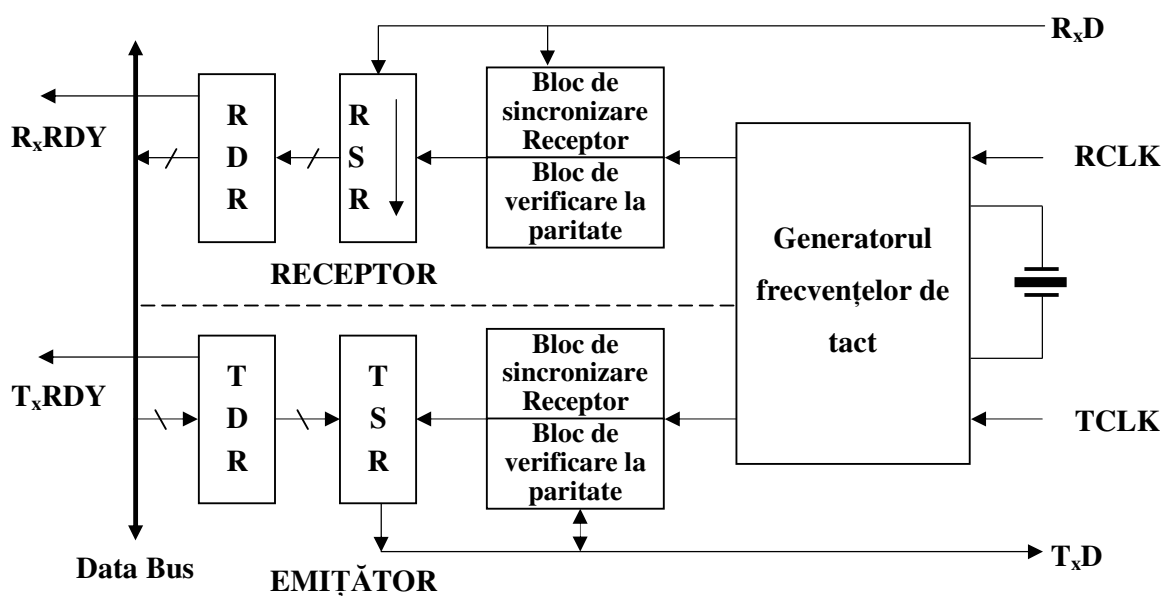
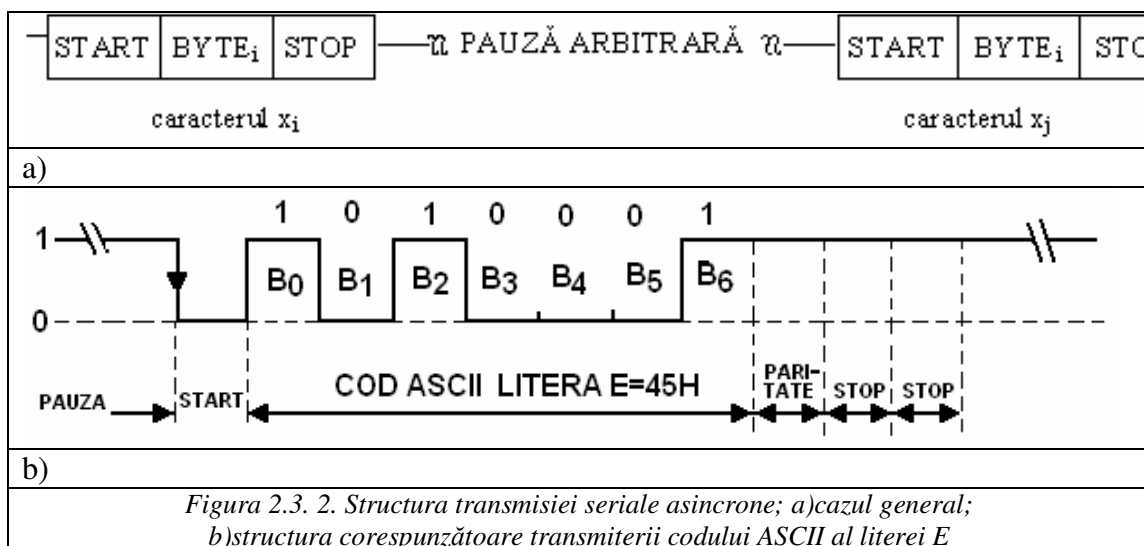
În cadrul protocolului dintre emițător și receptor sunt stabilite condițiile de comunicare și anume: viteza de transmisie ($1/T$), numărul de biți pe caracter, dacă există sau nu bit de paritate și numărul biților de *STOP*.

Recepția este declanșată pe frontul negativ al impulsului de *START*. Receptorul pornește cu acest front o bază de timp proprie (oscilatorul local) care va furniza momentele de eșantionare a semnalului recepționat, decalate cu $T/2$ față de frontul negativ al impulsului de tact. Recepția se termină după achiziționarea tuturor biților și se repetă pentru fiecare caracter transmis. Transmisia asincronă se utilizează de obicei la viteze mici și în funcție de viteză

Structuri hardware si algoritmi specifici microsistemelor EMBEDDED

rezultă o distanță maximă la care se poate transmite. De exemplu pentru 19200 Bit/s se pot asigura transmisii până la cca. 30m.

Schema de principiu a unei unități de comunicație serie, specializată în transmisii seriale asincrone, (*Universal Asynchron Receiver Transmitter*) este prezentată în Figura 2.3. 2.



- **Emisia**

Datele generale de procesor (sau a altor surse de informație) pe magistrala de date (Data Bus) se depun în registrul *TDR* (*Transmitter Data Register*). Datele sunt transferate apoi în registrul *TSR* (*Transmitter Shift Register*) în care se face, de fapt, conversia paralel-serie a datelor și se adaugă bitul de *START*. Datele sunt generate apoi bit cu bit începând

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

cu bitul de *START* pe ieșirea T_xD . În paralel cu transmisia biților de informație, aceștia sunt preluați și de generatorul de paritate care va calcula bitul de paritate al secvenței și îl va adăuga biților de informație. Secvența se termină cu adăugarea biților de *STOP*. Partea de emisie generează și un semnal de stare T_xRDY (*Transmitter Ready*) prin care indică faptul că datele au fost transferate din registrul *TDR* în registrul *TSR* și că se poate încărca un nou octet.

- **Recepția**

Datele recepționate bit cu bit pe intrarea R_xD se înscriu serie în registrul *RSR* (**R**eciever **S**hift **R**egister) și în paralel în blocul de verificare la paritate. În cazul în care paritatea este corectă și s-a detectat și numărul de biți de *STOP* stabiliți prin protocol, se dă comanda de transfer a biților de informație din registrul *TSR* în registrul *RDR* (**R**eciever **D**ata **R**egister). Semnalul de stare R_xRDY (*Receiver Ready*) indică faptul că în registrul *TDR* de la recepție se află un cuvânt de cod ce poate fi preluat de către procesor. Schema mai este prevăzută și cu un bloc de generare a frecvențelor de tact (Boundrate generator) folosit pentru a genera frecvența de tact necesară. Impulsurile de tact necesare se obțin fie pornind de la un oscilator de cuarț local, fie, în lipsa acestuia, trebuie aplicate unității de comunicație serie, tacte din exterior, pe intrările *TCLK* (*Transmitter Clock*) pentru emisie respectiv pe *RCLK* (*Receiver Clock*) pentru recepție.

2.3.3 Comunicația serie sincronă

Transmisia serială sincronă este utilizată pentru transmiterea unor blocuri mari de date. Exemple tipice de utilizare ar fi comunicațiile cu discurile magnetice (*FD* sau *HD*) respectiv comunicațiile la mare distanță prin *MODEM* (aplicații *INTERNET*). Unitățile de comunicație serială sunt în general circuite integrate complexe, multifuncționale care acoperă tot domeniul comunicațiilor seriale cum ar fi de exemplu un controller (*USART Universal Synchron Asynchron Receiver Transmitter*).

În comunicația serială sincronă există două protocoale de bază și anume:

- Protocol orientat pe caracter (*Character Oriented Protocol*).
- Protocol orientat pe bit (*Bit Oriented Protocol*)

Funcțiunile de bază pentru un protocol sunt:

- stabilirea și întreruperea legăturii între două stații;
- asigurarea integrității mesajelor (prin detecția erorilor, cerere de retransmisie și confirmări de recepție);
- identificarea transmițătorului și a receptorului;
- asigurarea unor funcții de comandă speciale.

2.3.4 Comunicația serie la microcontrolerele INTEL 51/31

Modul 0 sincron pe 8 biți cu rată fixă de transmisie

Schema bloc a circuitului de comunicație serială este prezentată în Figura 5.18. Datele sunt transmise pe terminalul R_xD ($P_{3,0}$ – pinul 0 al portului 3), iar tactul pe terminalul T_xD ($P_{3,1}$). Rata de transmisie este fixă și egală cu $f_{osc}/12$.

Modul de funcționare:

Emisia



UNIUNEA EUROPEANĂ



MINISTERUL MUNCII, FAMILIEI ȘI
PROTECȚIEI SOCIALE
AMFOSDRU



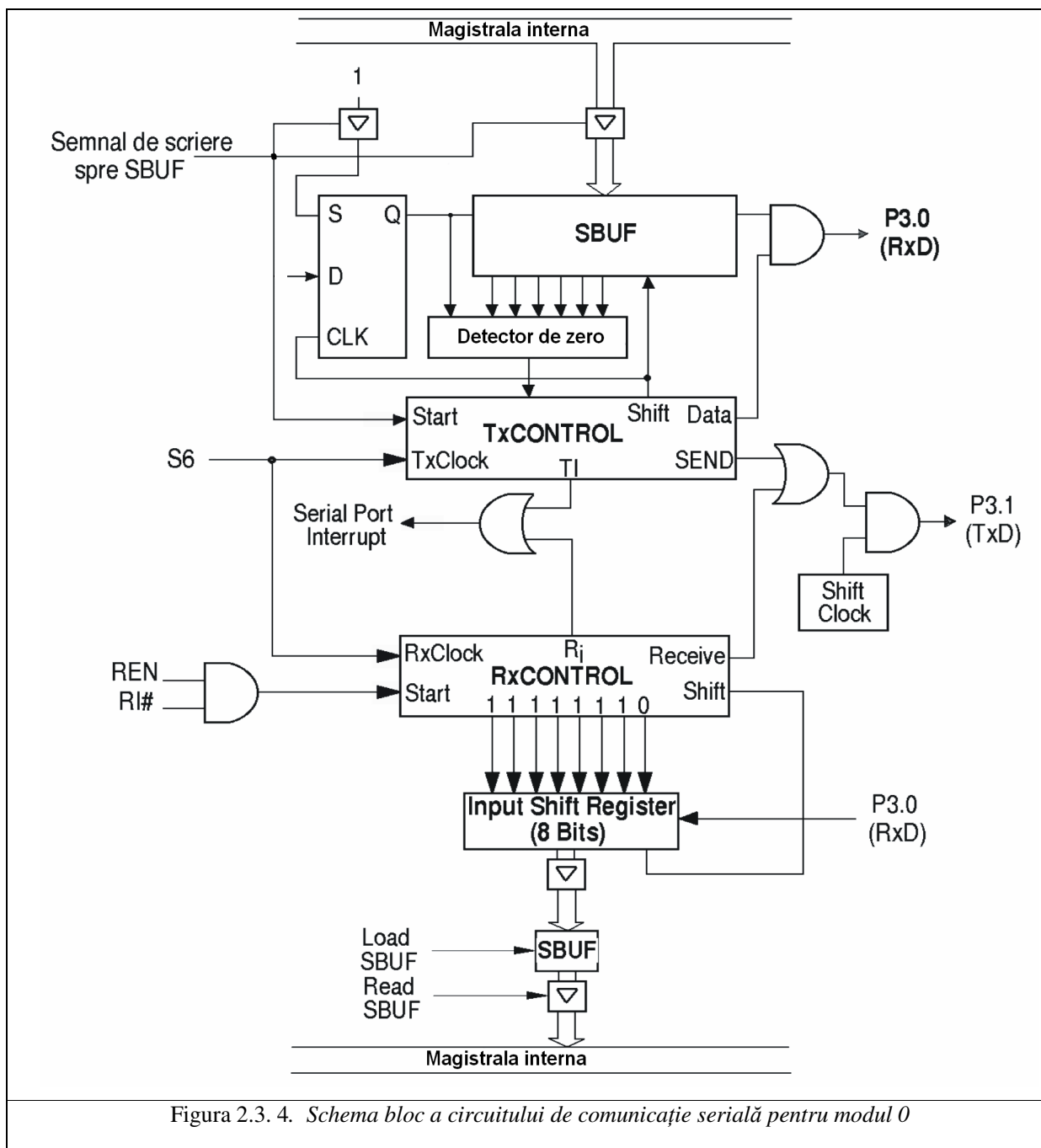
FONDUL SOCIAL EUROPEAN
POSDRU
2007-2013



INSTRUMENTE STRUCTURALE
2007-2013

Structuri hardware si algoritmi specifici microsistemelor EMBEDDED

Inițializarea comunicării se face în S_6P_2 (vezi Figura 2.3. 4) printr-o comandă de scriere în registrul *SBUF* (Convertor paralel-serie), concomitent cu aceasta se înscrie în celula *D* starea “1” și se comandă circuitul de control *TxControl*. După un ciclu mașină, se activează semnalul *SEND* care va deschide atât ieșirea de date RxD , cât și ieșirea de tact TxD .



Pe toată durata transmisiei se generează tactul TxD , cu comenzile S_3P_1 și S_6P_1 , precum și tactul de deplasare, cu S_6P_2 . Înaintea ultimului tact registrul *SBUF* conține informația 000000D₇. Circuitul de control sesizează această stare, generează încă un ultim impuls de tact și

Structuri hardware si algoritmi specifici microsistemelor EMBEDDED

setează un bit pentru o cerere de întrerupere. Diagramele temporale care ilustrează funcționarea în modul 0 sunt prezentate în *Figura 2.3. 5*.

Recepția

Pentru conversia serie–paralel se folosește registrul de deplasare (*RD*) de intrare, încărcat inițial cu *11111110*. În ritmul tactului, se citește semnalul de intrare în momentele *S₅P₂* și se înscriu datele în acest registru. Când zeroul ajunge în ultima poziție, circuitul de control *RxControl* recunoaște această stare, mai generează un impuls de tact, iar apoi transferă conținutul registrului *RD* în registrul *RM SBUF* (vezi *Figura 2.3. 5*).

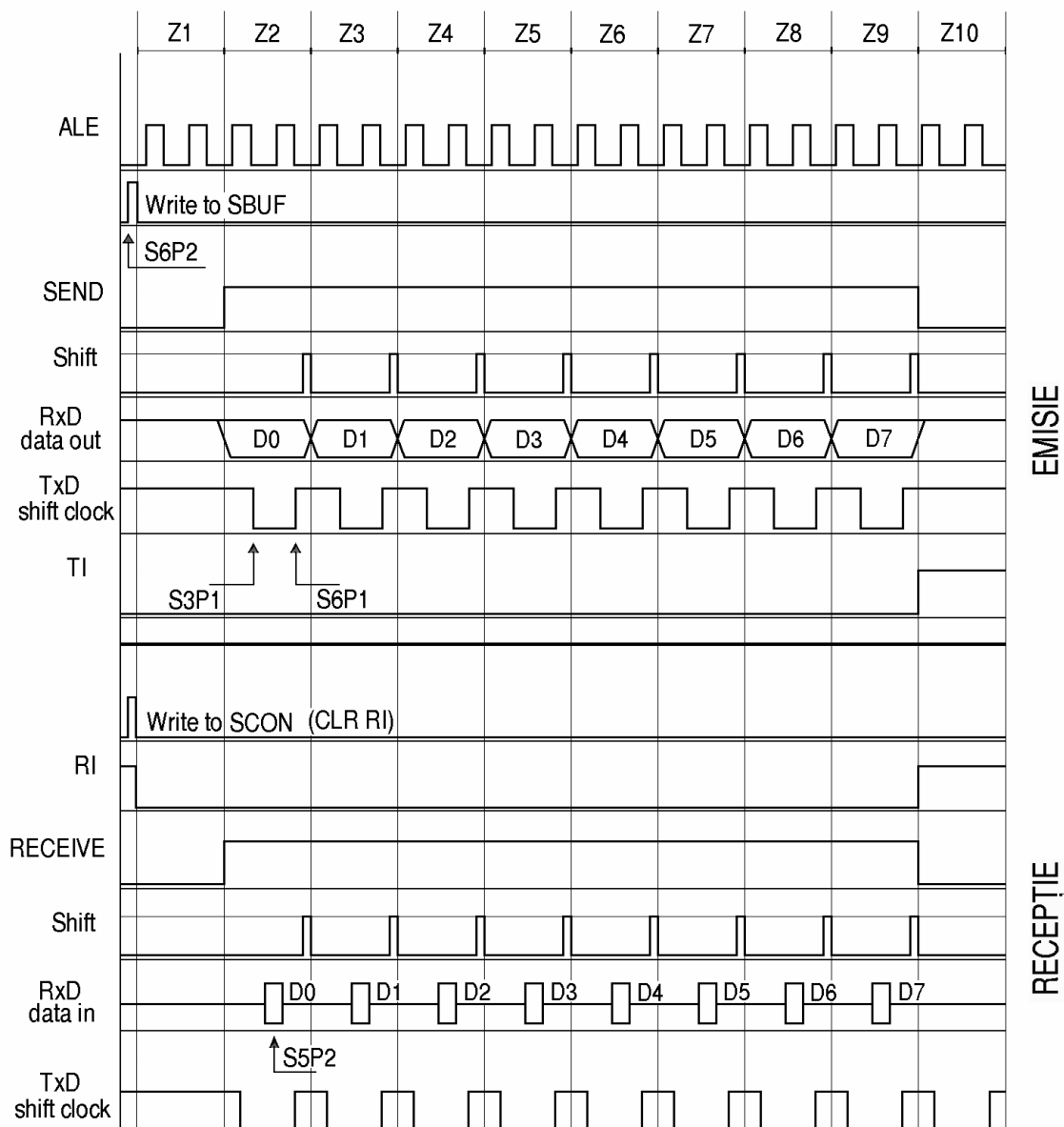


Figura 2.3. 5. Diagrame temporale ce ilustrează funcționarea în modul 0

Structuri hardware si algoritmi specifici microsistemelor EMBEDDED

genera impulsul de *START*. Recunoașterea stării $0000001D_7$ se face de către circuitul *TxControl*, care va mai genera un impuls pentru emisia lui D_7 și va seta bitul pentru cererile de întreruperi. Semnalul *SEND* va mai rămâne activ încă un ciclu mașină ca să încadreze octetul transmis cu bitul de *STOP* (vezi Figura 2.3. 6).

Recepția

Frontul negativ al impulsului de *START* determină *reset*-area *Timer*-ului. Tactul de recepție, ca și cel de emisie se obțin printr-o divizare cu 16 a semnalului overflow a *Timer*-ului 1. Cu biții 7, 8 și 9 ai tactului superior se testează semnalul recepționat și pe baza unui algoritm de decizie majoritar se înscrie 1 sau 0 în registrul de deplasare (de 9 biți) de la recepție (încărcat inițial cu *1FFH*).

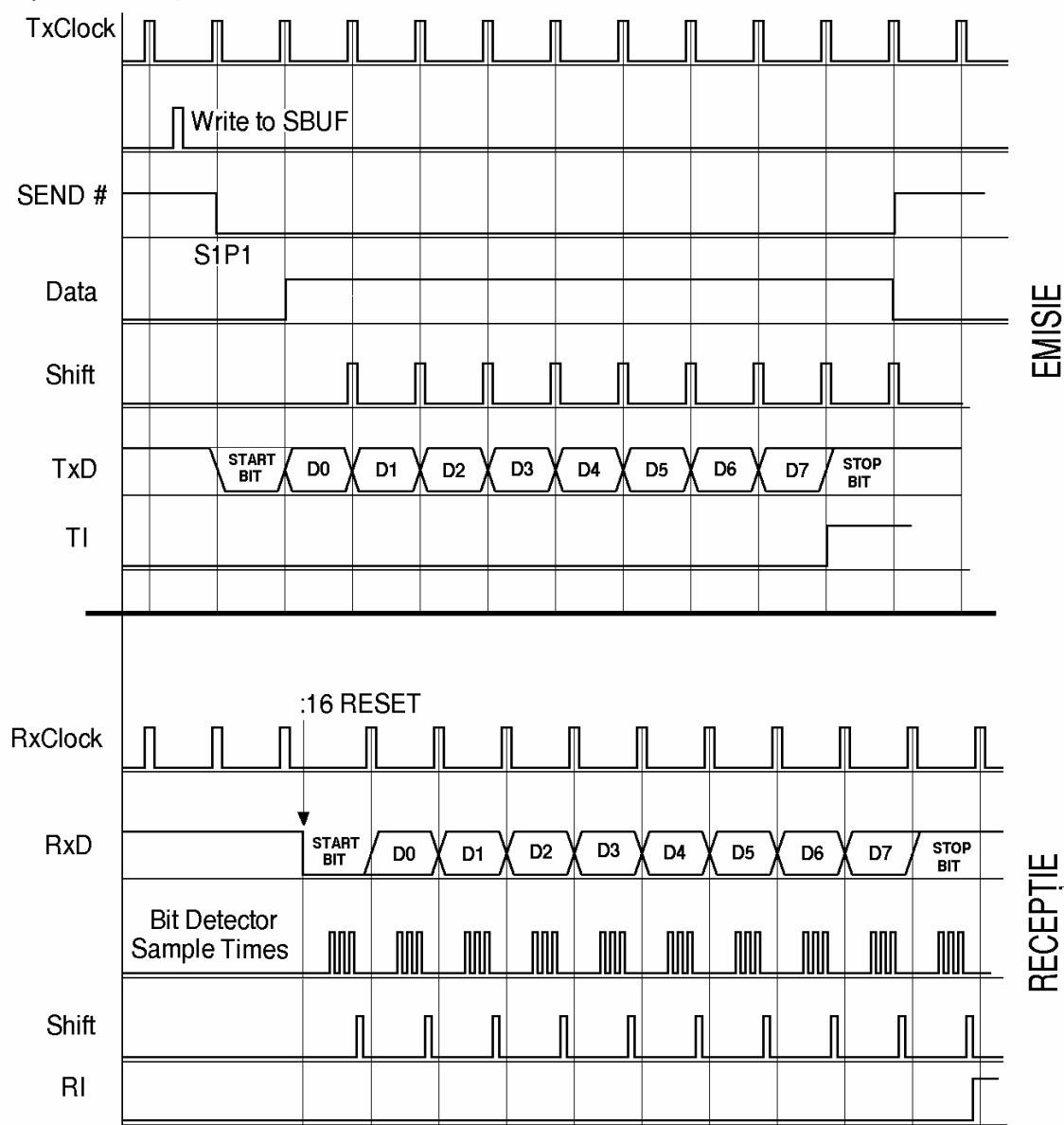


Figura 2.3. 7. Diagrame temporale ce ilustrează funcționarea în modul 1

Structuri hardware si algoritmi specifici microsistemelor EMBEDDED

Când ultima unitate ajunge în poziția din stânga a registrului, se va mai genera un ultim tact pentru a încălca și bitul de *STOP*, după care, în anumite condiții (se verifică corectitudinea recepției), informația utilă este transferată în *SBUF* (vezi Figura 5.20. și Figura 5.21.).

2.3.5 Regimul de comunicație multiprocesor

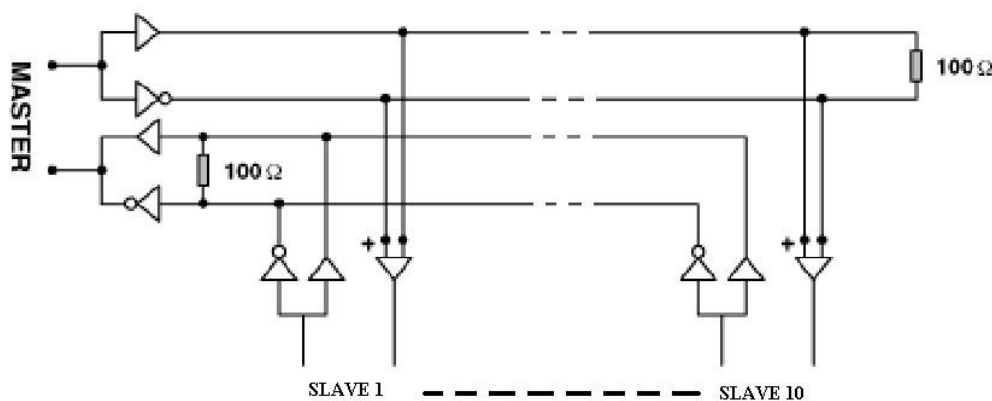
Modurile de lucru 2 și 3 sunt potrivite pentru a organiza o comunicație între mai multe controlere. Cu ajutorul celui de al nouălea bit de date memorat în registrul *SCON* pe poziția *RB8* poate fi generată o cerere de întrerupere dacă acest bit este la valoarea unu logic ($RB8=1$) și dacă este setat regimul de lucru în comunicație multiprocesor $SM2=1$.

În cazul în care un controler de pe post de *MASTER* dorește să transmită date unui alt controler aflat în postura de slave, trimite la început un cuvânt de adresă pentru a selecta respectivul controler *slave*. Diferența dintre un octet de adresă și unul de date îl dă tocmai cel de al nouălea bit. Un octet de adresă va avea cel de al nouălea bit de valoare unu logic iar un octet de date va avea cel de al nouălea bit de valoare zero logic.

Dacă bitul SM_2 este setat ($SM_2=1$) atunci valoarea de unu logic al celui de al nouălea bit va declanșa o întrerupere la toate microcontrolerele de tip *slave*. În cadrul rutinei de servire a întreruperii, fiecare controler de tip *slave*, poate decide dacă următoarele date îi sunt sau nu destinate. Controlerul adresat își va reseta bitul SM_2 fiind în felul acesta pregătit ca să preia octetul de date ce urmează. Controlerele *slave* neadresate vor lăsa bitul lor SM_2 setat și își va continua activitatea

2.3.6 Specificații la interfața serială RS422 si RS 485

RS422 este un standard diferențial care permite transmiterea datelor la distanțe mult mai mari decât RS232. Impedanta de intrare ridicată a dispozitivelor RS422, combinată cu capacitățile sporite ale ieșirii permit o conexiune cu pînă la 10 noduri.



Alt avantaj RS422 este utilizarea de cai de recepție și de transmisie separate pentru care nu este necesar controlul direcției. Orice procedură de handshaking necesară între dispozitive se realizează sau software (control XON/XOFF) sau hardware (pereche separată de semnal).

Parametru	Condiții	Min	Max	Unitate
Tensiune de ieșire în gol			± 10	V
Tensiune de ieșire în sarcină	Sarcină de 100Ω	+2 -2		V

Structuri hardware si algoritmi specifici microsistemelor EMBEDDED

Impedanța de ieșire	A la B		100	Ω
Curentul de scurtcircuit la ieșire	Intre ieșire si masa		150	mA
Timp de creștere la ieșire	Sarcina de 100 Ω		10	% din durata unui bit
Tensiunea de mod comun	Sarcina de 100 Ω		± 3	V
Senzitivitate receptor	$V_{cm} < \pm 7$ V		± 200	mV
Gama tensiunii de mod comun la receptor		-7	7	V
Impedanța de intrare receptor		4		k Ω
Tensiunea diferențială la receptor	Operațional		± 10	V

Dispozitivele RS422 si RS485 se confunda adesea; unele se presupun a fi versiunea duplex a celorlalte. Diferențele de natura electrica sunt totuși datorate gamei tensiunilor de mod comun si impedanțelor de intrare ale receptorilor, astfel ca se utilizează in aplicații diferite. Dispozitivele RS485 îndeplinesc toate specificațiile RS422, după cum se observa din tabelul următor, astfel ca pot fi folosite in aplicații RS422, dar invers nu este posibil. Gama tensiunii de mod comun pentru RS485 este de la -7V la +12V, mult mai mare decât gama -3V la +3V permisa pentru RS422. De asemenea, impedanța de intrare minima a receptorilor este 4k Ω pentru RS422 si 12 k Ω pentru dispozitivele RS485.

Parametru	Conditii	Min	Max	Unitate
Tensiune de iesire in gol		+1.5 -1.5	+6 -6	V
Tensiune de iesire in sarcina	Sarcina de 100 Ω	+1.5 -1.5	+5 -5	V
Curentul de scurtcircuit la iesire	Intre iesire si masa		250	mA
Timp de crestere la iesire	Sarcina de 54 Ω si 50pF		30	% din durata unui bit
Tensiunea de mod comun	Sarcina de 54 Ω		± 3	V
Senzitivitate receptor	-7V < V_{cm} < +12V		± 200	mV
Gama tensiunii de mod comun la receptor		-7	12	V
Impedanta de intrare receptor		12		k Ω

Structuri hardware si algoritmi specifici microsistemelor EMBEDDED

Permitind utilizarea de cabluri lungi, dispozitivele RS485 au devenit un standard popular in aplicații tip point-of-sale, industriale si de telecomunicații. Gama sporita a tensiunii de mod comun permite linii mai lungi si o impedanță de intrare mai mare per nod, astfel ca se pot conecta mai multe noduri la bus.

2.3.7 Topologii de rețea

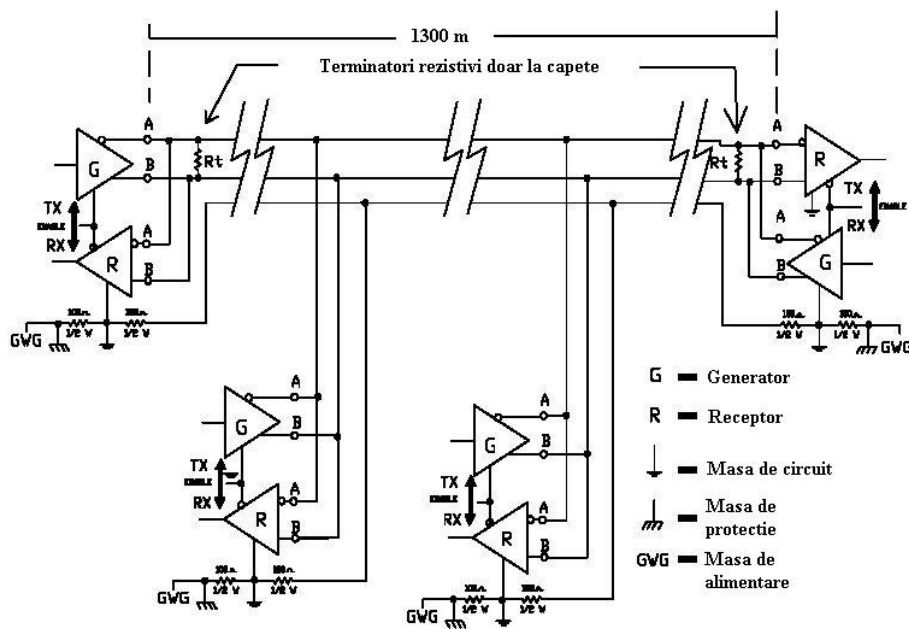
Configurația rețelei nu este definita in specificațiile RS422 sau RS485. In majoritatea cazurilor proiectantul poate astfel utiliza o configurație optimizata cerințelor fizice ale sistemului.

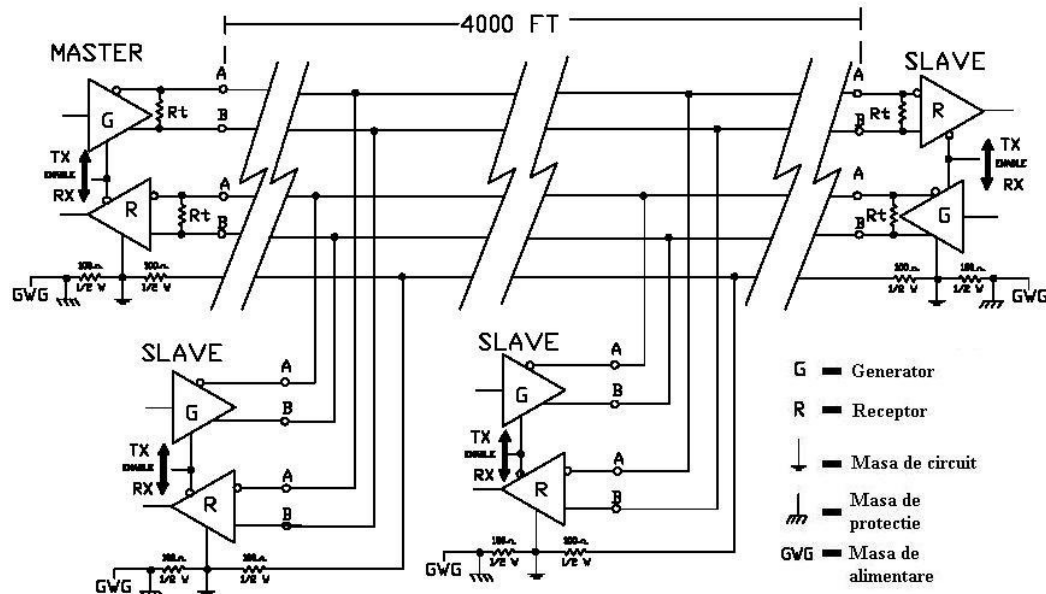
Sisteme pe doua sau patru fire

Sistemele RS422 necesita o pereche de fire pentru fiecare semnal, deci o pereche pentru transmisie, una pentru receptie si perechi aditionale pentru fiecare semnal de control utilizat (daca este cazul). Capabilitatile tristate ale sistemelor RS485 permit utilizarea unei singure perechi de fire pentru a partaja semnalele de transmisie si de receptie la comunicatii semiduplex. Aceasta configuratie „pe doua fire“ (in realitate trebuie folosit si al treilea, semnalul de masa) reduce costul cablarii.

Dispozitivele RS485 pot fi configurate intern sau extern pentru configuratii pe doua sau patru fire. Dispozitivele RS485 configurate intern pe doua fire ofera conexiunile A si B, marcate uneori si „-“ si „+“.

Dispozitivele configurate pentru comunicatii pe patru fire dispun de conexiuni A si B atat pentru receptie cit si pentru transmisie. Utilizatorul poate realiza configuratia in doua fire prin conectarea impreuna a liniilor de transmisie cu cele de receptie, astfel ca dispozitivele avind configuratie in patru fire ofera proiectantului flexibilitatea cea mai mare. In orice caz, semnalul de masa trebuie sa fie conectat in sistem, pentru a tine tensiunea de mod comun la receptor intr-o gama sigura de functionare. Circuitul de interfata poate functiona si fara semnalul de masa, dar cu pretul sacrificarii fiabilitatii si a imunitatii la zgomot. Figurile urmatoare ilustreaza conexiunile in cazul sistemelor pe doua si respectiv patru fire.





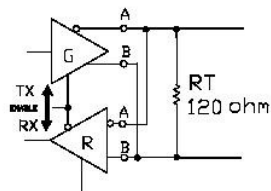
2.3.8 Adaptarea liniilor

Rezistoarele de terminatie se utilizeaza pentru a adapta impedanta nodului cu impedanta liniei de transmisie utilizate. In cazul neadaptarii impedantelor semnalul transmis nu este complet absorbit de sarcina si o anumita fractiune este reflectata inapoi in linia de transmisie. Daca impedantele sursei, liniei de transmisie si a sarcinii sint egale atunci aceste reflexii nu apar. Rezistoarele de terminatie cresc sarcina etajelor driver, sporesc complexitatea instalarii, modifica cerintele de incarcare si fac modificarea sistemului mai complicata.

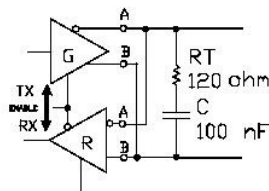
Decizia utilizarii adaptarii trebuie bazata pe studiul lungimii cablului si a ratei de transmisie a datelor din sistem. Ca regula generala, daca propagarea liniei de date este mult mai mica decit durata unui bit, nu este necesara adaptarea. Regula aceasta presupune ca reflexiile se vor atenua in citeva deplasari inainte si inapoi in linie. Intrucit unitatea UART receptoare va esantiona data la mijlocul duratei bitului, este important ca nivelul semnalului in acel moment sa fie stabil.

Exista mai multe metode de adaptare a liniilor de date. Metoda recomandata este adaptarea paralel. Aceasta metoda presupune conectarea in paralel cu liniile A si B ale receptorului a unei rezistente pentru a adapta impedanta caracteristica a liniei specificata de producatorul cablului (120 Ohmi este o valoare uzuala). Aceasta valoare descrie impedanta intrinseca a liniei de transmisie si nu depinde de lungimea liniei. O rezistenta de adaptare sub 90 Ohmi nu se recomanda a fi utilizata. Rezistoarele de adaptare trebuie plasate doar la extremitatile liniei si nu mai mult de doua intr-un sistem care nu utilizeaza repetoare. Acest tip de adaptare adauga incarcare masiva in curent continuu si poate supraincarca tipurile de convertoare RS232 la RS485 alimentate din port. Alt tip de adaptare este cea in curent alternativ, prin adaugarea unei capacitati reduse in serie cu rezistenta de adaptare, astfel eliminind efectul incarcarii in curent continuu. Dezavantajul metodei este dependenta inalta a selectiei capacitatii de proprietatile sistemului. Figura alaturata prezinta cele doua modalitati de adaptare a liniilor

Structuri hardware si algoritmi specifici microsistemelor EMBEDDED



Terminatie paralel



Terminatie cuplaj AC

Bibliografie:

1. Istvan Sztojanov, Sever Pașca, Elisabeta Buzoianu, Aplicații hardware și software cu microcontrolerul PIC12F675, Editura Cavallioti, ISBN 978-973-7622-54-9, Bucuresti 2008
2. Istvan Sztojanov, Alexandru Vasile, Elisabeta Buzoianu, Sever Pașca, *Programarea microcontrolerelor din familia Intel, Aplicații practice hardware cu 80C552*, Editura Man-Dely, ISBN 973-85681-5-3, București 2004.
3. <http://vega.unitbv.ro/~romanca/EmbSys/>
4. <http://facultate.regielive.ro/cursuri/electronica/>
5. www.microcip.com
6. Andrei Drumea, Teza de doctorat, UPB 2009



UNIUNEA EUROPEANĂ



MINISTERUL MUNCII, FAMILIEI ȘI
 PROTECȚIEI SOCIALE
 AMFOSDRU



FONDUL SOCIAL EUROPEAN
 POS DRU
 2007-2013



INSTRUMENTE STRUCTURALE
 2007-2013