

5b. Interfețe seriale. RS232 și circuite programabile



Cuprins modul

5b.1. Standardul RS232

5b.2. Circuit de interfață programabil

A. Descrierea circuitului

B. Funcționarea circuitului

C. Programarea circuitului

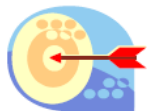
5b.3. Modificarea nivelului de tensiune

Cuprins



Introducere

Modulul “**Interfețe seriale. RS232 și circuite programabile**” continuă prezentarea interfețelor seriale care a început în modulul 5a cu aspecte generale de codare și de formare a cuvintelor și cadrelor seriale. În acest modul se va prezenta exemplul concret al standardului de transmisie serială RS232, care este implementat în majoritatea microcontrollerelor. Un exemplu de circuit de interfață programabil și modul de programare vor completa imaginea acestei interfețe seriale. La sfârșit se vor menționa câteva aspecte legate de modificarea nivelului de tensiune.



Obiective

După parcurgerea acestui modul studenții vor cunoaște interfața RS232 și implementarea ei cu un circuit de interfață programabil. Studenții vor putea să:

- Conceapă o interfață serială asincronă cu o interfață specializată pentru a o conecta pe magistrală;
- Programeze o interfață serială RS232 din structura unui microcontroller.

Obiective specifice:

1. Învățarea noțiunii de protocol cu aplicare la transferul de date prin interfețe;
2. Învățarea programării la nivel de bit a interfețelor seriale și paralele;
3. Cunoașterea unor tipuri de transmisii seriale și interfețe seriale ca structură, protocol și interfețe tipice.
4. Înțelegerea noțiunilor prin exemplificări practice



Durata medie de studiu individual

Durata medie de studiu individual este de 2 ore.

5b.1. Standardul RS232

EIA (*Electronics Industries Association*) împreună cu TIA (*Telecommunication Industry Association*) au realizat standarde pentru interfețe seriale. Standardul utilizat în prezent pentru conectarea microcontrollerelor la un calculator gazdă este EIA/TIA RS232-E. Valorile tensiunilor admisibile pentru fiecare nivel logic sunt date în figura 5.10. Valorile maxime admisibile sunt +25V și -25V.

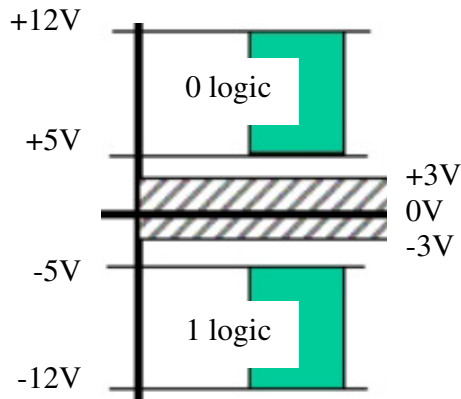


Figura 5.10. Nivelele de tensiune ale valorilor logice în RS232

Nivelele mai mari de tensiune asigură o margine de zgomot mai mare și prin urmare o imunitate mai mare la perturbații. Distanța maximă de transfer este de 15m la un debit de informație de 115Kbps.

Cele mai importante semnale de interfață în standardul RS232 sunt:

Tabelul 1

Semnal	Sens	Semnificație
RxD	←	recepție date
TxD	→	emisie date
RxC	←	ceas de recepție
TxC	←	ceas de emisie
RTS	→	Request to Send (cerere de emisie)
CTS	←	Clear to Send (gata de emisie)
DTR	→	Data Terminal Ready (terminal de date gata?)
DSR	←	Data Set Ready (terminal de date gata)

Termenii Half Duplex (HDX) și Full Duplex (FDX) se referă la simultaneitatea transferului de date în ambele sensuri între două echipamente. Dacă există flux de date atât într-un sens cât și în celălalt, dar nu simultan, atunci transferul este HDX. Dacă există flux de date într-un sens și în celălalt în același timp, atunci transferul este FDX. În industrie se folosește pentru HDX termenul TWA (Two Way Alternate) și pentru FDX termenul TWS (Two Way Simultaneous).

O legătură punct la punct conectează două dispozitive, iar o legătură multipunct conectează mai mult de două dispozitive. Aceste legături sunt ilustrate în figura 5.11. În legătura punct la punct, unul din dispozitive este emițător și celălalt este receptor (stânga). În legătura multipunct un singur dispozitiv este emițător și dintre celelalte, la un moment dat, unul sau mai multe dispozitive sunt receptoare (dreapta). Receptorul este activat printr-un mecanism de adresare specific interfeței.

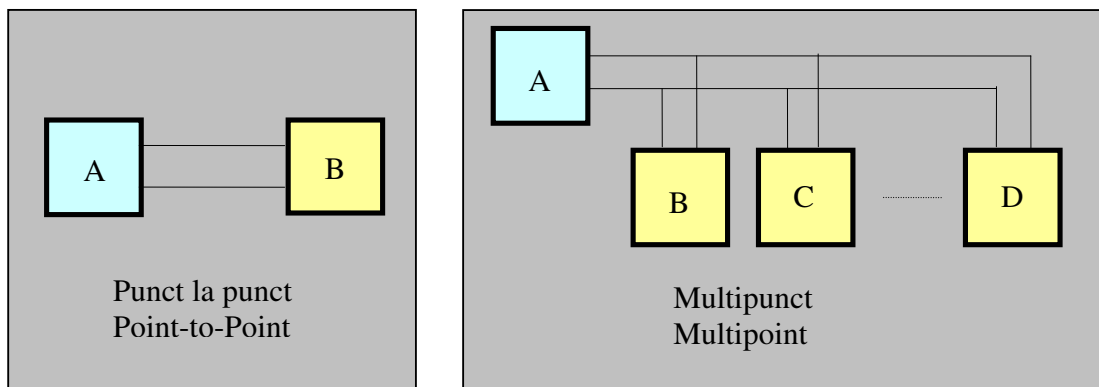


Figura 5.11. Legătură punct la punct și multipunct

Din punctul de vedere al acestor precizări, interfața serială RS232 este o interfață punct la punct, Full Duplex.

O legătură completă RS232 între două sisteme de calcul utilizează toate semnalele principale din Tabelul 1 și este dată în figura 5.12.

Datele circulă prin liniile TxD și RxD în mod FDX sau HDX, depinde de capacitatea circuitelor seriale de interfață. Perechile de semnale RTS /CTS și DTR /DSR au rolul de a implementa un protocol hardware de comunicație. Unul dintre sistemele conectate solicită un transfer prin semnalul RTS sau DTR, iar celălalt prin CTS sau DSR confirmă disponibilitatea receptorului de a primi date. Tactul de recepție și cel de emisie pot fi diferite dare de cele mai multe ori ele sunt egale și provin de la un genrator extern. Transmisia serială RS232 poate fi cu tact standard și în acest caz cele două generatoare de tact generează un tact precizat în standard și în foile de catalog a circuitelor. Dacă

tactul este generat la un singur sistem și este transmis prin linia de transmisie se obține o transmisie serială cu transmiterea tactului, care poate asigura un debit mai mare de informație. În transmisia RS232 cele două sisteme trebuie să aibă masă comună, deci conexiunea trebuie completată cu o linie de masă (GND).

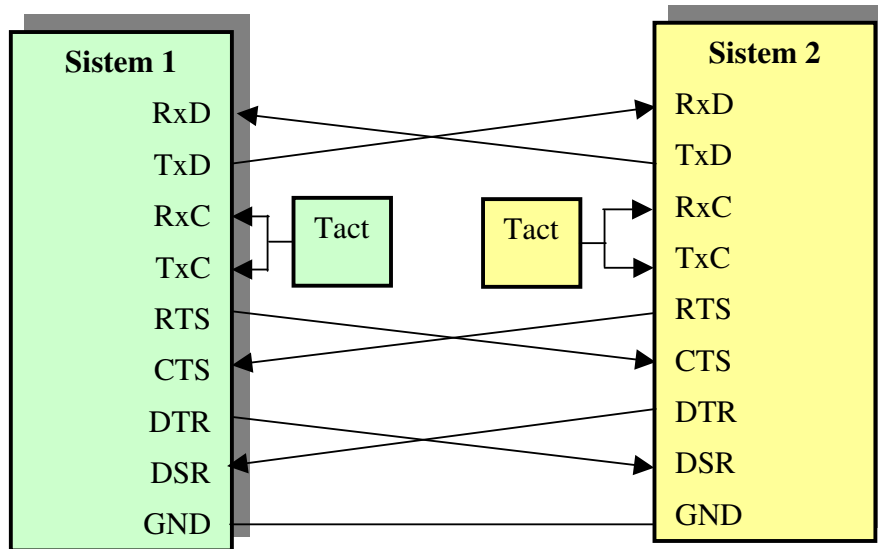


Figura 5.12. Legătură RS232 completă



Conexiunea între cele două sisteme se face cu 7 fire. Avantajul interfețelor seriale este numărul mic de linii de conectare. Nu este aici o contradicție?



Ba da, de aceea de cele mai multe ori nu se folosesc toate liniile de protocol.

Dintre cele două perechi de linii de protocol RTS /CTS și DTR /DSR se poate folosi doar una, scăzând astfel numărul de linii de conexiune de la 7 la 5. Protocolul hardware de comunicație se numește DTR sau RTS în funcție de perechea utilizată. Pentru transfer poate fi implementat și un protocol software care se numește XonXoff. Acest protocol înseamnă transmiterea pe linia TxD și recepția pe RxD a unor coduri, unul care semnifică “liber” (Xon) și unul care semnifică “ocupat” pentru Xoff. În această situație numărul de linii de conexiune scade la 3, figura 5.13.

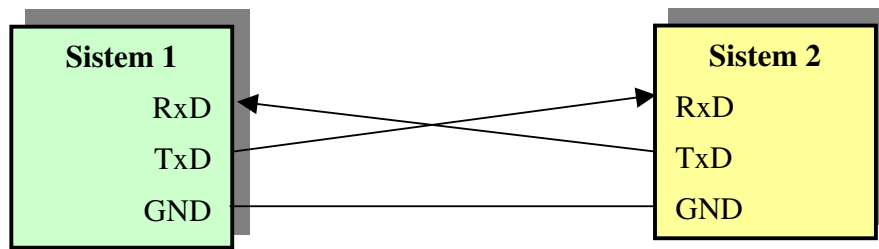


Figura 5.13. Transmisia RS232 pe 3 fire cu protocol software

Tactul de transmisie este standard dar evoluția echipamentelor a necesitat mărirea vitezei de comunicație. De aceea s-au definit viteze standard de comunicație iar circuitele de interfață serială prelucrează prin divizare tactul standard pentru a obține mai multe viteze de comunicație standard, selectabile software. Vitezele de comunicație standard sunt (în Baud): 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

5b.2.Circuit de interfață programabil

A.Descrierea circuitului

Ca exemplu de circuit programabil de interfață serială RS232 a fost ales circuitul Intel 8251, care a fost primul circuit conceput în acest scop de Intel. Circuitul se numește circuit USART (Universal Synchronous Asynchronous Receiver Transmitter) pentru că poate lucra atât în mod asincron cât și sincron. Dacă la interfețele paralele primul circuit programabil realizat de Intel a fost utilizat o lungă perioadă de timp, circuitul de interfață serială I8251 a fost înlocuit după scurt timp de interfețe mai evolute. Alegerea acestui circuit s-a datorat valorii educative, pentru că realizează atât mod asincron cât și sincron spre deosebire de circuitele mai noi care realizează doar modul asincron al interfeței RS232. Un alt motiv este simplitatea circuitului I8251 și modul simplu de programare.

Schema bloc a circuitului este dată în figura 5.14.

Semnalele de interfață cu microprocesorul gazdă au aceeași semnificație ca la circuitul de interfață paralelă:

- RD se execută cu ciclul de citire de la un port sau de la memorie. Acest pin se conectează la semnalul de magistrală IOR;
- WR se execută cu ciclul de scriere la un port sau în memorie. Acest pin se conectează la semnalul de magistrală IOW;

- C/D, Comenzi / Date, cu 0 arată transfer de date, cu 1 transfer de comenzi. Se conectează de regulă la linia cel mai puțin semnificativă de adresă;
- CLK tact de magistrală pentru comanda operațiilor interne în interfață;
- RESET este o linie care comandă inițializarea circuitului prin ștergerea informației din toate registrele;
- CS este o linie care selectează circuitul. Formării acestui semnal i se dedică un modul de studiu ulterior;
- D0-D7 magistrala de date a gazdei, 8 linii bidirecționale;

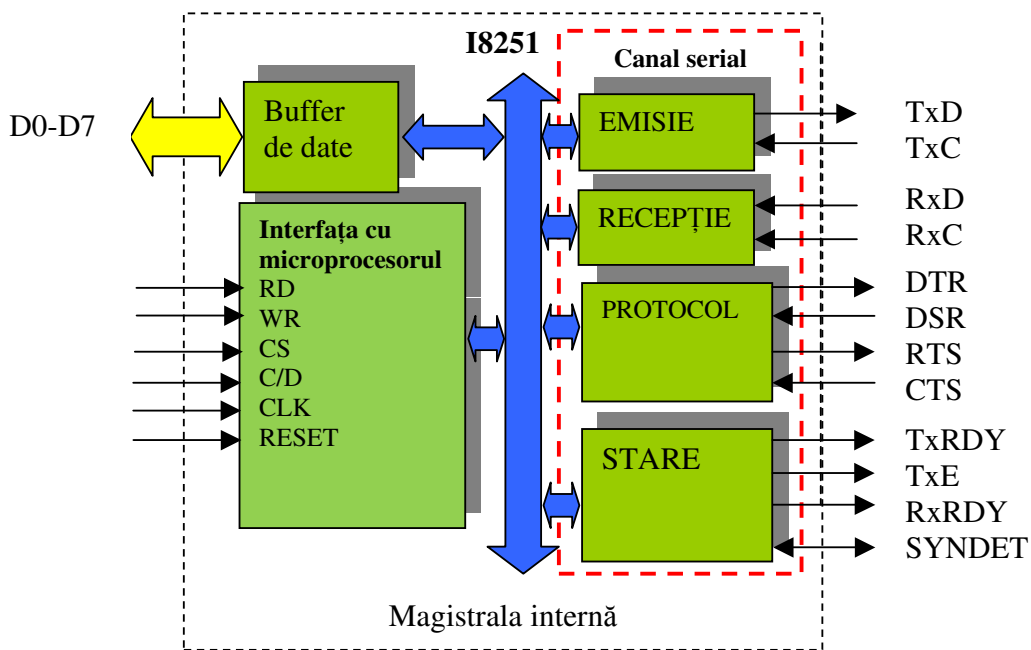


Figura 5.14. Schema bloc a circuitului programabil de interfață serială I8251

Semnalele de conectare cu linia RS232 sunt cele standard, descrise la 5b.1. În plus față de acestea există câteva semnale care au rolul de a informa procesorul gazdă de starea circuitului. Aceste semnale sunt:

- TxRDY transmițător pregătit pentru a primi date de pe magistrala D0-D7;
- RxRDY receptorul a recepționat un caracter care este gata de a fi trimis pe magistrala D0-D7;
- TxR transmițător gol;
- SYNDET un semnal cu sens dublu, interfața anunță că a detectat sincronizarea sau este anunțată că un circuit extern a detectat sincronizarea.

Aceste semnale pot fi utilizate ca cereri de întrerupere pentru procesor.

B.Funcționarea circuitului

Circuitul are ca și funcție principală conversia paralel serie și serie paralel. Un cuvânt care trebuie transmis paralel este scris în circuit în format paralel de către procesor pe liniile D0-D7, este serializat și în mod asincron i se adaugă biți de START, STOP și de paritate. Serializarea se face cu tactul TxC. Este transmis apoi serial pe linia TxD, cu un protocol hardware activând și citind perechile RTS/CTS, DTR/DSR sau un protocol software XonXoff. După transmisia cuvântului se activează semnalul TxRDY pentru ca procesorul să poată trimite alt cuvânt.

La recepția în mod asincron informația serială care vine pe linia RxD este citită serial cu tactul RxC începând cu primul front care este recunoscut ca bit de START. Se verifică paritatea, apoi sunt eliminați biții de START și STOP și se face conversia serie paralel. Circuitul activează semnalul RxRDY pentru ca procesorul să știe că poate citi caracterul în format paralel pe liniile D0-D7.

În mod sincron la recepție circuitul poate lucra în două moduri, cu sincronizare internă sau cu sincronizare externă. Cu sincronizare internă la recepție citește continuu datele de pe linia RxD și le compară cu un cuvânt de 8 biți de sincronizare. Acest cuvânt este standardizat, este folosit la cadre de date cu codare ASCII și marchează începutul unui cadru de date. Acest mod de funcționare în care circuitul urmărește sincronizarea se numește mod *Hunt*. În acest mod circuitul nu transmite date spre procesor. În momentul recunoașterii cuvântului de sincronizare, datele sunt citite serial de pe linia RxD, sunt transformate în format paralel, se verifică bitul de paritate și sunt transmise spre procesor pe liniile D0-D7. La detectarea sincronizării se activează semnalul SYNDET. Semnalul RxRDY se activează la fiecare cuvânt recepționat.

În mod sincron la recepția cu sincronizare externă, un circuit extern se ocupă cu recunoașterea cuvântului de sincronizare. Când este recunoscut un astfel de cuvânt se activează către circuitul de interfață serială semnalul SYNDET și circuitul începe să citească serial date de pe linia RxD, sunt transformate în format paralel, se verifică bitul de paritate și sunt transmise spre procesor pe liniile D0-D7.

În mod sincron la transmisie un cadru de date este transferat octet după octet în format paralel pe liniile D0-D7 către circuit, circuitul inserează cuvântul de sincronizare, calculează și inserează bitul de paritate, serializează cuvintele și le transmite pe linia TxD. Când a terminat de trimis un cuvânt semnalizează procesorului prin activarea semnalului TxRDY.

În mod sincron cuvântul de sincronizare poate avea un octet sau doi octeți.

C.Programarea circuitului

Primul cuvânt de comandă trimis în circuit cu linia C/D=1 comandă modul de lucru și principalii parametri de funcționare, astfel:

Dacă D1=0 și D0=0 modul de lucru este sincron, la orice altă combinație modul este asincron.

Semnificația biților din primul cuvânt de comandă în mod asincron

D7	D6	biți de STOP
0	0	invalid
0	1	1 bit
1	0	1 ½ bit
1	1	2 biți

D5	D4	paritate
X	0	dezactivat
0	1	paritate pară
1	1	paritate impară

D3	D2	lungime caracter
0	0	5 biți
0	1	6 biți
1	0	7 biți
1	1	8 biți

D1	D0	Tact de transmisie
0	0	invalid
0	1	tact/64
1	0	tact/16
1	1	tact

Semnificația biților din primul cuvânt de comandă în mod sincron

D7=0 cuvântul de sincronizare are doi octeți, D7=1 cuvântul de sincronizare are 1 octet

D6=0 sincronizare internă, D6=1 sincronizare externă

D5	D4	paritate
X	0	dezactivat
0	1	paritate pară
1	1	paritate impară

D3	D2	lungime caracter
0	0	5 biți
0	1	6 biți
1	0	7 biți
1	1	8 biți

D1=0, D0=0 pentru selectarea modului sincron

Dacă s-a selectat modul sincron cu sincronizare externă atunci după primul cuvânt de comandă nu mai urmează alt cuvânt de programare specific modului sincron. Dacă s-a selectat modul sincron cu sincronizare internă cu un octet de sincronizare atunci urmează un al doilea cuvânt de programare care conține octetul de sincronizare. Dacă s-a selectat modul sincron cu sincronizare internă cu doi octeți de sincronizare atunci urmează al doilea și al treilea cuvânt care conțin acești doi octeți.

Al doilea cuvânt de comandă și cuvântul de stare

Structura cuvântului al doilea de comandă este dată în stânga și a cuvântului de stare (citit cu semnalul C/D=1) este dată în dreapta.

Al doilea cuvânt de comandă		Cuvânt de stare	
Bit	Semnificație	Bit	Semnificație
D7	Intrare în mod <i>Hunt</i>	D7	DSR
D6	Reset intern	D6	SYNDET
D5	RTS	D5	FE (Frame Error)
D4	Anularea tuturor erorilor din registrul de stare	D4	OE (Overrun Error)
D3	Transmisie caracter BREAK	D3	PE (Parity Error)
D2	Activare recepție	D2	TxE
D1	DTR	D1	RxRDY
D0	Activare emisie	D0	TxRDY

Primul cuvânt de comandă programează parametric comunicației, iar al doilea cuvânt de comandă declanșează execuția unei operații: transmisia, recepție, modul de detectare a sincronizării etc. Transmisia unui caracter BREAK înseamnă că linia de transmisie este trecută în stare SPACE și nu mai este posibilă transmisia informațiilor pe linie. O utilizare posibilă a acestui caracter BREAK este la detectarea automată a vitezei de transmisie. Lungimea caracterului, prin urmare durata menținerii liniei în stare SPACE poate da informații receptorului despre viteza de transmisie, pentru ca acesta să-și poată adapta tactul de recepție. Cu al doilea cuvânt de comandă se pot genera semnalele de protocol DTR și RTS dacă protocolul folosit este cel hardware. Dacă se folosește protocolul software aceste linii nefolosite se pot utiliza de exemplu pentru semnalizarea stării circuitului.

Cuvântul de stare conține starea liniilor TxE, RxRDY, TxRDY și SYNDET pentru a putea fi citite de procesor în cazul în care transferul este programat. În cuvântul de stare se poate citi semnalul de protocol DSR și se pot identifica erorile care au apărut la transmisie: eroarea de paritate PE când bitul de paritate transmis nu corespunde cu cel generat la receptor, eroarea de sincronizare FE când nu s-a recepționat un bit de STOP corespunzător și eroarea de suprascriere (OE) când s-a recepționat un cuvânt înainte ca cel anterior să fie citit.

5b.3.Modificarea nivelului de tensiune

În figura 5.10 se arată că nivelele de tensiune RS232 sunt între 3 și 15V, iar din figura 5.14 se vede că circuitul interfață serială nu are circuite de modificare de nivel și fiind în tehnologie TTL are nivele între 0 și 5V. Prin urmare, semnalele generate și recepționate de circuitul interfață serială trebuie convertite ca nivel în semnale RS232. Pentru aceasta, cel mai cunoscut și utilizat circuit este MAX232. De cele mai multe ori, modulele echipate cu interfață serială sunt alimentate doar de la 5V, așa încât circuitul MAX232 conține în interior două convertoare DC/DC pentru a forma +12V și -12V din 5V. Schema bloc internă a circuitului MAX232 este dată în figura 5.15 (stânga):

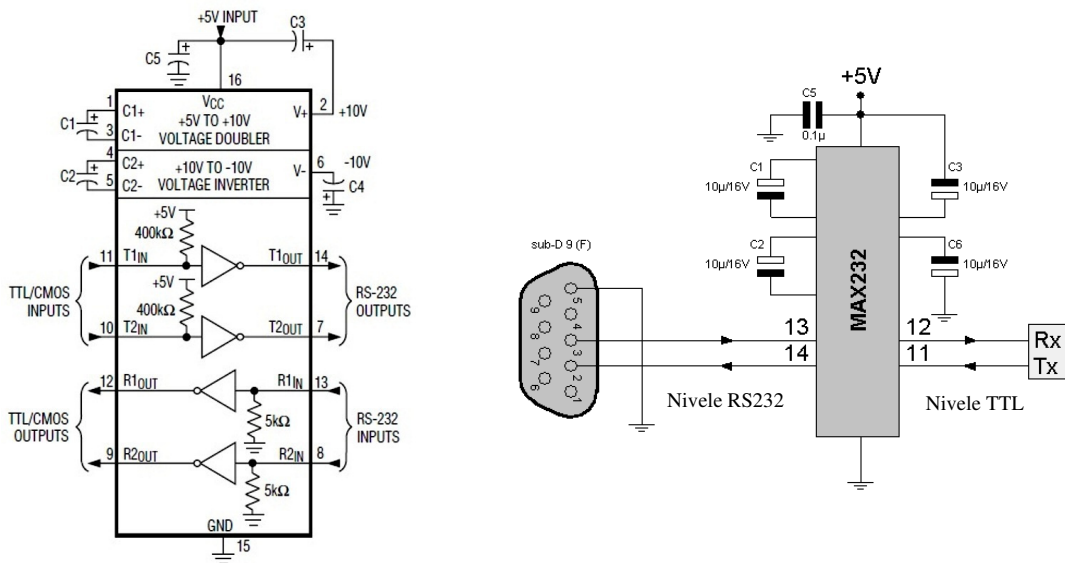


Figura 5.15. Schema bloc internă a circuitului MAX232

În figura 5.15 în dreapta se poate vedea o schemă electrică tipică de utilizare a unui circuit MAX232 în cazul unei legături seriale RS232 cu protocol software, la care comunicația se face pe două fire TxD și RxD, și GND.

În figura 5.16. se poate vedea o diagramă de timp vizualizată cu un osciloscop pentru transmiterea serială a unui caracter. Jos este reprezentat semnalul la nivele TTL iar sus se observă semnalul cu amplitudine mai mare (RS232). Se poate observa că polaritatea la RS232 este inversată de MAX232.



Figura 5.16. Diagrama transferului serial RS232 a unui caracter (jos la nivel TTL, sus la nivel RS232)



La examinarea de final de curs este nevoie ca studenții să memoreze poziția fiecărui bit de programare?



Nu. Întrebarea de la test va fi asociată cu un text care va conține toate datele necesare rezolvării.



Se poate proiecta o aplicație cu datele furnizate de acest modul de învățare?



Nu, pentru că modulul este didactic și nu conține detalii privind variația în timp a semnalelor, valorile lor logice în care semnalele sunt active, etc. Pentru proiectare este nevoie de consultarea foilor de catalog.



Cele mai multe calculatoare PC nu mai au interfață serială RS232. Ce se poate face în acest caz pentru conectarea unui dispozitiv RS232?



Se poate utiliza un convertor USB RS232.



Este complicat de a asigura o conexiune serială între un microcontroller și un PC prin interfața serială?



Din figura 5.17 se poate vedea că o astfel de conexiune este simplă.

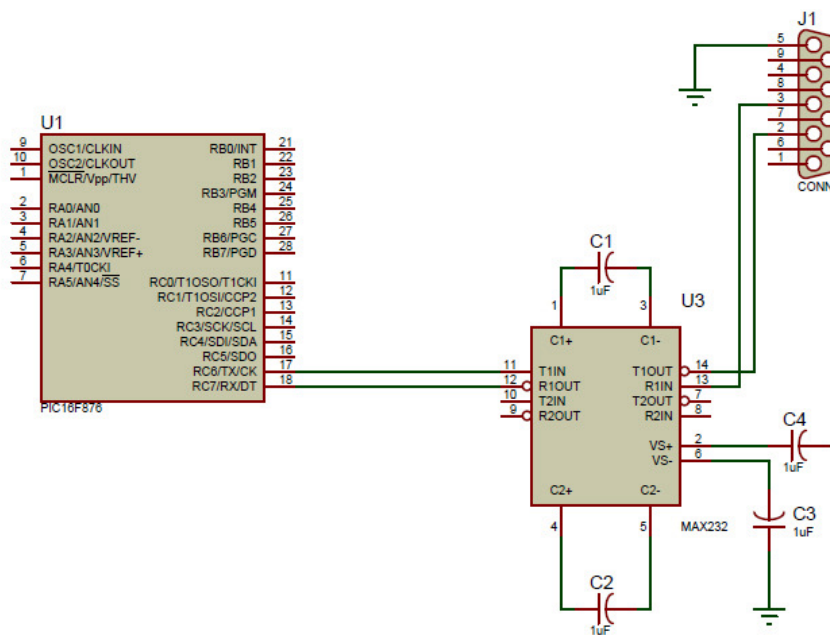


Figura 5.17. Conectarea unui microcontroller la un PC prin interfața serială cu MAX232, sursa <http://creativeelectron.net/blog/2009/10/pic-serial-communiton-in-pic-microcontroller-1/>



Rezumat

Modulul “**Interfețe seriale**” a fost împărțit în două. În această parte a doua este descris standardul serial RS232 ca nivele de tensiune, semnale principale și protocol de comunicație. Este descris apoi circuitul interfață serială programabilă Intel 8251 cu schema bloc, funcționare și programare. Se justifică necesitatea circuitelor de conversie de nivel și se descrie sumar circuitul MAX232. O diagramă de timp vizualizată cu un osciloscop marchează sfârșitul celor două module dedicate interfețelor seriale. Câteva dintre modulele următoare vor fi dedicate altor interfețe seriale, de exemplu Ethernet, IEEE 1394, IrDA, SATA etc.



Bibliografie

1. C. Gerigan, P. Ogruțan, *Tehnici de interfațare*, Ed. Transilvania Brașov, 2000, 315p., ISBN 973-9474-94-2, pag. 69-93, online la: <http://vega.unitbv.ro/~ogrutan/ti/cap5.pdf>
2. RS-232 online la: <http://en.wikipedia.org/wiki/RS-232>
3. RS232 detalii, online la: <http://academic.evergreen.edu/projects/biophysics/tech/notes/electron/serial.htm>
4. I8251 online la: <http://www.electronics.dit.ie/staff/tscarff/8251usart/8251.htm>
5. <http://creativeelectron.net/blog/2009/10/pic-serial-communication-in-pic-microcontroller-1/>

Test de autoevaluare



**1.O legătură HDX este realizată între 2 interfețe seriale RS232.
Se transmit date în același timp pe linia TxD și se
recepționează date pe linia RxD.**

R

adevărat

fals

I. vezi pagina 3

**1.Comunicația RS232 este gestionată cu un protocol soft care
include coduri diferite trimise pe liniile DTR /RTS și
recepționate pe liniile DSR / CTS.**

R

adevărat

fals

I. vezi pagina 3 și 4

**3.Semnalul SYNDET este generat de interfața serială I8251
dacă:**

R

(a) un circuit extern de sincronizare a detectat
aparitia unui octet de sincronizare

(b) circuitul I8251 a detectat aparitia unui octet de
sincronizare

(c) în mod asincron s-a recepționat un cod Xon

(d) la recepția fiecărui caracter

I. vezi pagina 7

4. Un circuit I8251 este conectat pe o magistrală de 8 biți de adrese și 8 biți de date, semnalul C/D fiind conectat la linia de adrese A0. Secvența de cod următoare:

R

```
MOV DX,0000H
```

```
MOV AL,55H
```

```
OUT DX,AL
```

Are ca efect:

- (a) trimiterea în 8251 a primului cuvânt de programare
- (b) trimiterea în 8251 a celui de al doilea cuvânt de programare
- (c) trimiterea octetului 55H în 8251 ca și octet de date și transmisia lui pe linia TxD
- (d) trimiterea octetului 55H în 8251 ca și octet de date

I. vezi pagina 6, 8 și 9

5. Programați circuitul 8251 pentru mod asincron cu viteza maximă de transfer, 1 bit de STOP, paritate pară, 8 biți/ caracter:

R

```
MOV DX, adresa port serial
```

```
MOV AL, H
```

```
OUT DX,AL
```

I. Vezi pagina 8. (biții care sunt indiferenți se pun pe zero)

6. Un circuit I8251 este conectat pe o magistrală de 12 biți de adrese și 8 biți de date la adresa 0000H, semnalul C/D fiind conectat la linia de adrese A0. Comandați circuitul să execute transmisia caracterului AAH cu parametri de comunicație: mod asincron cu viteza minimă, 1 bit de STOP, fără paritate, 8 biți/ caracter:

R

```
MOV DX, [ ]H
MOV AL, [ ]H
OUT DX,AL
MOV AL, [ ]H
OUT DX,AL
```

I. Vezi paginile 6,8, 9. (biții care sunt indiferenți se pun pe zero)

R

Răspunsuri corecte:

1. fals, HDX nu înseamnă transmisie și recepție în același timp, pagina 3
2. fals, codurile se trimit pe linia TxD și se recepționează pe RxD, pagina 3 și 4
3. b, SYNDET este activat doar la modul sincron și doar la sincronizare internă, realizată de I8251, pagina 7
4. d, adresa 0000H corespunde la C/D=0, deci octet de date. Nu există al doilea cuvânt de programare care să comande transmisia, deci răspunsul c nu este corect, paginile 6,8 și 9.
5. 5FH, 01011111 binar, pagina 8
6. 0001H pentru ca C/D=1 și se trimite primul cuvânt de comandă 4DH, apoi al doilea 01H care validează transmisia. Paginile 6,8,9.