6 Comunicație serială . Protocolul RS232

6.1 Scopul lucrării:

În această lucrare de laborator se va studia protocolul de comunicație serială RS232 şi se va implementa în CVI o comunicație bazată pe acest protocol. Se recomandă utilizarea aplicației realizate în lucrările de laborator precedente.

6.2 Generalități

6.2.1 Magistrale seriale de comunicație

Magistralele seriale se utilizează ca suport pentru transferul de informații între calculatoare sau între componentele autonome ale unui sistem de calcul (sisteme embedded, un sistem embedded şi PC etc.). Caracteristica principală a oricărei magistrale seriale este transmisia secvențială, bit cu bit, a informațiilor, folosindu-se un număr redus de semnale (linii de comunicație). În contrast, o magistrală paralelă permite transferul simultan al mai multor biti (8, 16, 32), folosind în acest scop mai multe linii de date. În principiu transmisia serială asigură o viteză de transfer mai redusă, în comparație cu transmisia paralelă, însă este mai economică (număr mai redus de linii de transmisie), iar distanța maximă de transfer este semnificativ mai mare.

Magistralele seriale pot fi clasificate după mai multe criterii :

a. după modul de sincronizare:
   - transfer sincron - se utilizează un semnal explicit de ceas (de sincronizare) pentru specificarea momentului în care un bit de dată este valid;
   - transfer asincron - nu se utilizează semnal de ceas, sincronizarea între unitatea emitentă şi cea receptoare se face în mod implicit pe baza structurii specifice a datei transmise (a se vedea standardul RS 232).
b. după lungimea blocului de date transmis:
   - transfer pe octet;
   - transfer pe bloc (număr mai mare de octeți).
c. după numărul de unități comunicante:
   - transfer serial de tip punct-la-punct : legatura se realizează între două echipamente;
   - transfer serial multipunct : legatura se realizează simultan între mai multe echipamente, din care la un moment dat unul transmite şi restul ascultă
d. după direcția de transfer:
   - transfer unidirecțional (într-un singur sens);
   - transfer bidirecțional sau « full duplex »(simultan în două sensuri);
   - transfer bidirecțional pe o singură linie sau « half duplex »(se transmite pe rând în cele două direcții).
e. după domeniul de utilizare:
   - magistrale de sistem - folosite pentru interconectarea componentelor unui microsistem (ex : microcontroller, memorii, convertoare A/D si D/A etc.)
   - canale de comunicație serială – folosite pentru interconectarea unor echipamente inteligente (ex : calculatoare, imprimantă , consolă) prin legătura punc-la-punct;
   - rețea de comunicație – folosită pentru asigurarea comunicației multipunct între un set de echipamente de calcul (observație : într-o accepțiune mai restrânsă rețelele de comunicație nu fac parte din clasa magistralelor seriale).

Transferul serial se realizează pe baza unui set de reguli care alcătuiesc protocolul de comunicație. Două echipamente care comunică pe o magistrală serială (canal serial) trebuie să

respecte acelaşi protocol şi aceeaşi parametri de transmisie (ex : viteza de transfer, mod de sincronizare, lungimea blocului de date, etc.).

Pentru a asigura interoperabilitatea între diferite echipamente realizate de diverşi producători, s-au definit o serie de standarde internaționale, care specifică :

- modul de transmisie a datelor (sincron/asincron);
- modul de structurare a datelor transmise (octet, bloc);
- viteza de transmisie;
- mecanismele de detecție şi corecție a eventualelor erori;
- tipul semnalelor folosite pentru transmisie (tensiune, curent, tensiune diferențiala, etc.);
- mecanismele de sincronizare a echipamentelor comunicante (ex : protocol XON/XOFF; sincronizare prin semnale explicite, etc.);
- tipul de conectori folosiți;
- natura şi parametrii fizici ai mediului de transmisie (ex : cablu bifilar torsadat, cablu coaxial, fibra optica, etc.);

Cele mai cunoscute standarde folosite pentru comunicația serială sunt : RS 232, RS 485, I2C, SPI si HDLC/SDLC. In continuare se prezintă câteva caracteristici mai importante ale standardului RS232.

6.2.2 Standardul RS232

Este cel mai cunoscut şi utilizat standard de comunicație serială asincronă. El a fost definit de mai multe organisme internaționale de standardizare sub diferite nume : IEC232, CCITT-V24, RS232C. Inițial standardul a fost conceput cu scopul de a permite conectarea unui terminal inteligent la un calculator central printr-o legătură telefonică. Standardul precizează interfața dintre un echipament de calcul (DTE- Data Terminal Equipment) şi adaptorul sau la linia telefonică (DCE - Data Circuit - terminating Equipment), cunoscut şi sub numele de modem (Modulator / Demodulator). Interfața permite comunicația serială bidirecțională intre cele două echipamente, şi este simetrică la cele doua capete ale liniei. Ulterior specificațiile acestei interfețe s-au folosit pentru a realiza legături seriale între diverse echipamente fără a se mai folosi un modem.

Principalele precizări ale standardului RS232 se referă la :

- modul de transmisie : serial asincron, bidirecțional (pe doua linii de date separate)
- codificarea informațiilor binare : prin nivele de tensiune sau curent (bucla de curent) :
  - 1 logic – (-3V … -15V);
  - 0 logic – (+3V…-15V).
- structura informației elementare transmise :
  - un bit de start (0 logic);
  - 5-8 biți de date.
  - 0-1 bit de paritate (paritate para sau impara);
  - 1-2 biți de stop (1 logic).

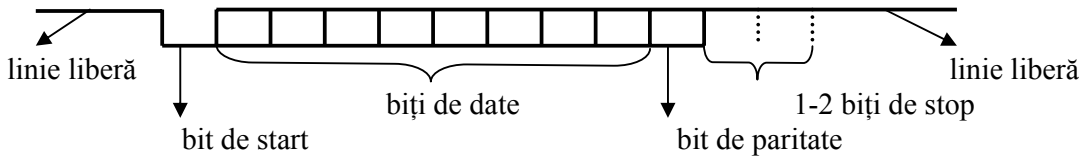

Figura 6.1 Structura unui caracter transmis conform standardului RS232

- semnale utilizate pentru transmisia de date şi pentru controlul fluxului de date (vezi tabelul 1);
- tipul de conectori folosiți (RK 25, mufa şi soclu) şi poziția semnalelor pe pinii conectorilor;

- modul de interconectare a semnalelor la cele doua capete ale unui cablu de transmisie;
- viteza de transmisie (110, 300, 600, 1200, 2400, 4800, 9600, 19200 bauds);
- reguli de control al fluxului de date (control hardware – protocolul DTR/DSR sau software - protocolul XON/XOFF);

In tabelul de mai jos s-a indicat numele si semnificația celor mai importante semnale definite de standardul RS232. De asemenea s-a indicat poziția acestor semnale pe un conector de 25 pini şi pe unul de 9 pini. Direcția este indicată intre calculator (DTE) şi modem (DCE).

Nume semnal	Semnificația/Funcția	Direcție DTE-DCE	Poziția pe con.RK25	Poziția pe con. RK9
RXD	Receive Data - recepție date	←	3	2
TXD	Transmit Data – transmisie date	→	2	3
GND	Masa digitala	--		5
DTR	Data Terminal Ready – terminal pregătit pentru transmisie	→	20	4
DSR	Data Set Ready – Pregăteşte dispozitiv pentru transmisie	←	6	6
RTS	Request To Send – Cerere de transmisie	→	4	7
CTS	Clear To Send – Pregătit pentru transmisie	←	5	8
RI	Ring – sonerie	←	22	9
CD	Carrrier Detect – detecție purtătoare	←	8	1

În cazul transmisiei seriale asincrone, sincronizarea între unitatea emitentă şi cea receptoare se realizează la începutul fiecărui caracter prin bitul de start (0 logic). De precizat că în repaus linia este in 1 logic. Citirea datelor se face secvențial, la jumătatea intervalelor de bit care urmează bitului de start. Protocolul asigură citirea corecta a datelor chiar si in cazul in care există mici diferențe (sub 2%) intre frecvența de emisie si cea de citire a datelor. Această sincronizare nu s-ar păstra în cazul în care lungimea datelor utile ar fi mai mare. Pentru controlul fluxului de date transmise se poate utiliza un protocol hardware sau unul software. În primul caz se utilizează semnale explicite (grupul de semnale DTR/DSR sau RTS/CTS) prin care unitatea receptoare poate să oprească temporar fluxul de date transmis. In acest fel se poate sincroniza frecventa de emisie a datelor la viteza de prelucrare a unitarii receptoare. A doua metoda nu utilizează semnale de control ; in schimb foloseşte un set de coduri speciale prin care poate să oprească (codul XOFF) sau sa repornească (codul XON) fluxul de date. Această metoda se poate utiliza numai la transmiterea unor date in codificare ASCII. La transmisia binara codurile de control ar putea sa fie prezente in datele de transmis.

În cazul în care se conectează două echipamente aflate la distantă mică (ex : în interiorul unei încăperi) se pot utiliza numai o parte din semnalele precizate in interfața RS232. In acest fel cablul de legătură devine mai ieftin si mai uşor de manipulat. In continuare se prezintă câteva configurații tipice de interconectare.

a. Transmisie unidirecțională, fără controlul fluxului de date

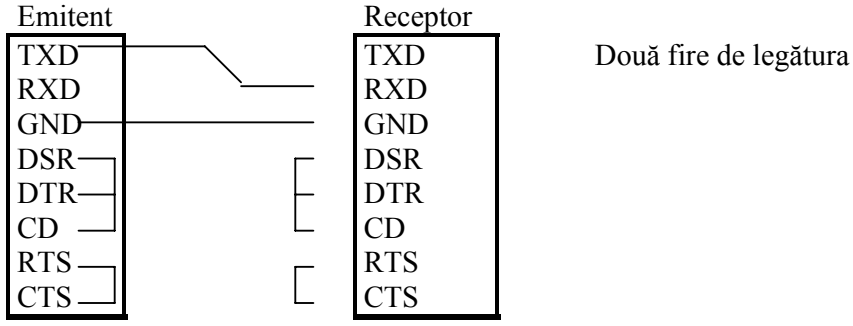

Figura 6.2 Transfer unidirecțional

b. Transmisie bidirecționala folosind protocolul XON/XOFF

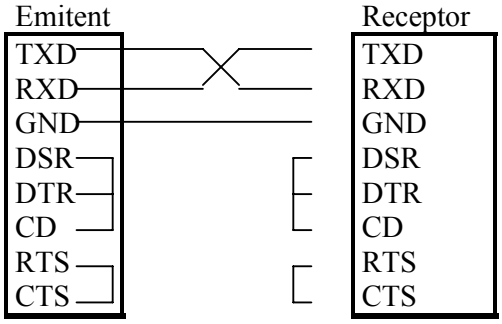

Figura 6.3 Transfer bidirecțional, protocol XON/XOFF

c. Transmisie bidirecționala folosind protocolul DTR/DSR

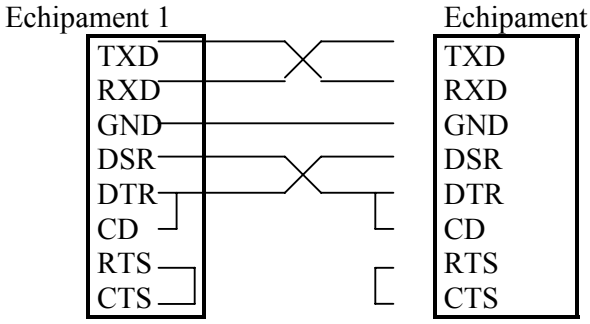

Figura 6.4 Transfer bidirecțional, protocol DTR /DSR

d. Transmisie bidirecționala folosind interfața completa RS232

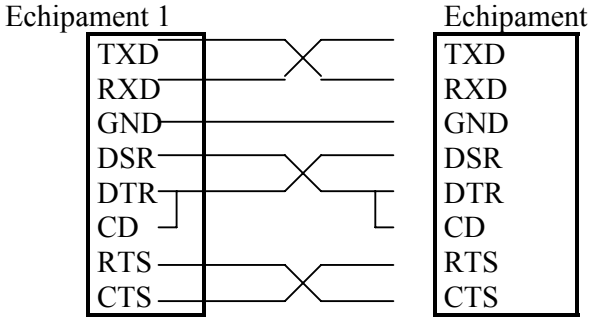

Figura 6.5 Transfer bidirecțional protocol DTR/DSR si RTS/CTS

Majoritatea calculatoarelor actuale dispun de cel puțin o interfață seriala. La calculatoarele de tip PC una din interfețele seriale este utilizată de obicei pentru a asigura legătura cu dispozitivul de indicare de tip Mouse.

Portul serial al unui calculator personal este **full duplex** însemnând că poate transmite date în timp ce recepționează. Acest lucru este posibil prin folosirea de linii separate pentru transmisie şi recepție.

Data bits – odată ce bitul de start a fost transmis, trasmițătorul trimite şi biții de date ce pot fi în număr de 5, 6, 7 sau 8. Atât transmițătorul cât şi receptorul trebuie să aibă setat acelaşi număr de biți de date. După ce biții de date au fost transmişi, unul, 1.5 sau doi **biți de stop** sunt transmişi. Acesta are valoare de „1" şi poate fi detectat corect chiar dacă ultimul bit de date are tot valoarea „1". Numărul de biți de stop poate fi:1, 1.5, 2.

Pe lângă sincronizarea dată de folosirea biților de start şi stop mai există un bit, numit bit de paritate, opțional, transmis împreună cu biții de date. Bitul de paritate permite într-o mică măsură

detecția erorilor ce pot apărea în timpul transmisiei. Pot fi alese paritățile even parity, odd parity, mark parity, space parity sau nici una.

Unitatea de măsură baud a fost numită după Jean Maurice Emile Baudot, un ofițer în serviciul de telegrafie fancez. El a fost primul care a creat un cod pe 5 biți pentru caracterele din alfabet la sfârşitul secolului 19. **Baud rate** se referă, în cazul conectării a două calculatoare, calculator – microsistem etc., la numărul de biți transferați pe secundă şi se poate spune că baud rate = bits per second. Dacă, în schimb, este conectat un modem, baudrate înseamnă de câte ori linia îşi schimbă starea.

Lungimea cablului, conform standardului, nu poate fi mai mare de 15,25 metri, dar această limită poate fi ignorată dacă este folosit un cablu de foarte bună calitate şi bine ecranat, funcție de baudrate:

#	Rată de transfer	Lungimea cablului ecranat (m)	Lungimea cablului neecranat (m)
1	110	1524	304
2	300	1219	304
3	1200	914	152
4	2400	609	152
5	4800	152	76
6	9600	76	30

6.3 Biblioteca RS232 a LabWindows/CVI

Mediul de dezvoltare LabWindows/CVI conține o bibliotecă de funcții pentru gestionarea comunicației seriale (open/close, input/output, control, status, callback, etc.):

Deschis / Închis

OpenComConfig - Deschide un port COM şi setează parametrii acestuia;
CloseCom – Închide un port COM;
OpenCom – Deschide un port COM folosind parametri impliciți;

Intrare / Ieşire

ComRd – Citeşte un număr specificat de octeți din bufferul de intrare şi îi stochează în bufferul dat ca parametru;

ComRdTerm – citeşte din bufferul de intrare până când este întâlnit codul ASCI al unui character dat ca parametru ori s-a citit un număr specificat de octeți (oricare din condiții apare prima);

ComRdByte – Citeşte un octet din bufferul de intrare al portului specificat ca parametru;

ComToFile – Citeşte din bufferul de intrare al portului dat ca parametru şi scrie datele în fişierul specificat de fileHandle dat de asemenea ca parametru;

ComWrt – Scrie un număr de octeți în bufferul de ieşire al portului specificat;

ComWrtByte – Scrie un octet în bufferul de ieşire al portului specificat;

ComFromFile – Citeşte din fişierul specificat şi scrie în bufferul de ieşire al portului dat ca parametru;

Control:

SetComTime – Setează limite de timeout pentru operații de intrare ieşire;

SetXMode – Validează sau invalidează software handshaking validând sau invalidând senzitivitatea XON / XOFF la transmisia şi recepția datelor.

SetCTSMode – Setează hardware handshaking.

FlushInQ – Şterge toate caracterele din bufferul de intrare al portului dat ca parametru.

FlushOutQ - Şterge toate caracterele din bufferul de ieşire al portului dat ca parametru.

Status

GetComStat – Returnează informații despre starea unui port serial;

GetComConnectionState – Returnează starea conexiunii unui port serial;

GetInQLen – Returnează numărul de caractere din bufferul de intrare a unui port;

GetOutQLen - Returnează numărul de caractere din bufferul de ieşire a unui port;

ReturnRS232Err – Returnează codul de eroare al celui mai recent apel de funcție din procesul curent;

GetRS232ErrorString – converteşte un cod de eroare returnat de una din funcțiile de mai sus într-un şir de caractere ce conține informații despre eroare respectivă;

Callback

InstallComCallback – instalează o funcție sincronă de callback pentru un port serial;

Pentru mai multe detalii se va consulta help-ul.

Funcțiile sunt astfel proiectate încât să vină în întâmpinarea programatorului, stabilirea şi controlul unei comunicații seriale fiind foarte uşor de realizat.

6.4 Teme

1. La proiectul realizat în lucrările de laborator precedente, adăugați controalele prezentate în figura 6.6 (şase controale de tip ring, un buton de tip toggle, trei butoane de comandă, o căsuță de introdus text şi două text box-uri).
2. Controalele adăugate pe interfață se vor activa/dezactiva în funcție de starea butonului Rețea.
3. Se va folosi un cablu serial null modem pentru a conecta două calculatoare între ele. Comunicația serială se va testa între cele două calculatoare.
4. Când se apasă butonul de tip toggle Connect, va trebui să se deschidă un port serial ce va avea configurațiile setate în interfață (se va utiliza funcția **OpenComConfig**). La a doua apăsare pe buton, se va închide portul serial (**CloseCom**).
5. Ataşați o funcție de callback pentru comunicația serială utilizând funcția **InstallComCallback,** astfel încăt această funcție de callback să se apeleze ori de câte ori este recepționat un caracter pe serială.
6. În funcția de recepție creată la punctul 5, preluați caracterul recepționat şi afişați-l în fereastra RECEIVE.
7. La apăsarea butonului Send, se va citi căsuța de introducere a textului şi se va transmite mesajul pe interfața serială. De asemenea, căsuța de text TRANSMIT va fi reactualizată iar căsuța de introducere a textului se va şterge.
8. Butoanele ClearRx şi ClearTx vor şterge conținutul ferestrelor RECEIVE si TRANSMIT.
9. Să se actualizeze ori de câte ori este necesar status-comunicației seriale (căsuța Status).
10. Să se transmită pe serială eşantioanele provenite de la generarea unui semnal prin interfață (sinusoidal, triunghiular, dreptunghiular, zgomot). ATENȚIE: eşantioanele reprezintă valori reale (double) iar pe intrerfața serială se poate trimite la un moment dat un singur octet.

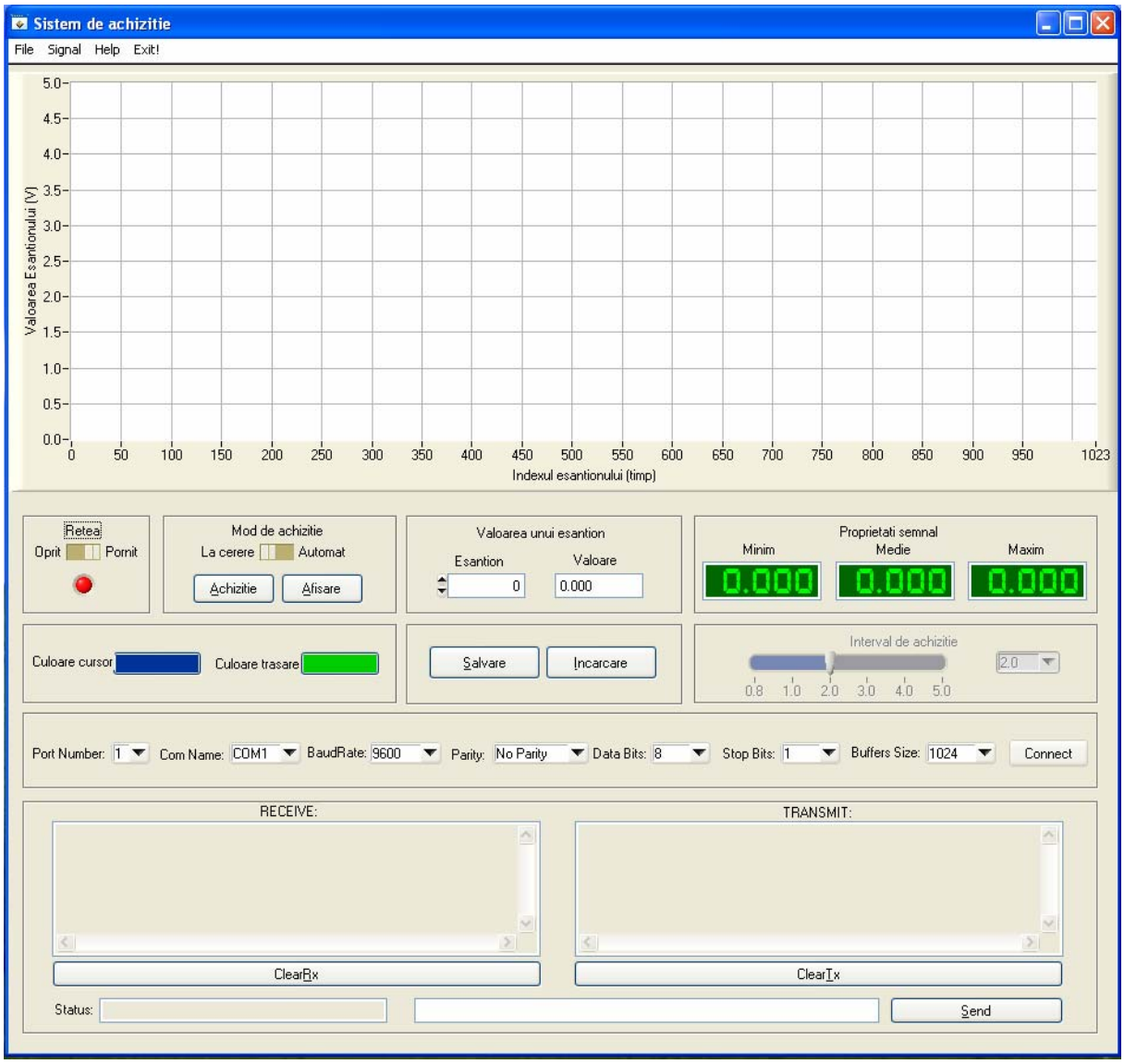
Sistem de achizitie
File Signal Help Exit!
Valoarea Esantionului (V)
Indexul esantionului (timp)
Retea
Oprit Pornit
Mod de achizitie
La cerere Automat
Achizitie
Afisare
Valoarea unui esantion
Esantion
Valoare
Proprietati semnal
Minim
Medie
Maxim
Culoare cursor
Culoare trasare
Salvare
Incarcare
Interval de achizitie
Port Number:
Com Name:
BaudRate:
Parity:
Data Bits:
Stop Bits:
Buffers Size:
Connect
RECEIVE:
TRANSMIT:
ClearRx
ClearTx
Status:
Send

Figura 6.6

Cuprins