

**Universitatea Tehnica a Moldovei**

**Facultatea Calculatoare, Informatica si Microelectronica**

**Departamentul Informatica si Ingineria Sistemelor**

**Disciplina:  
Interfete de Comunicare**

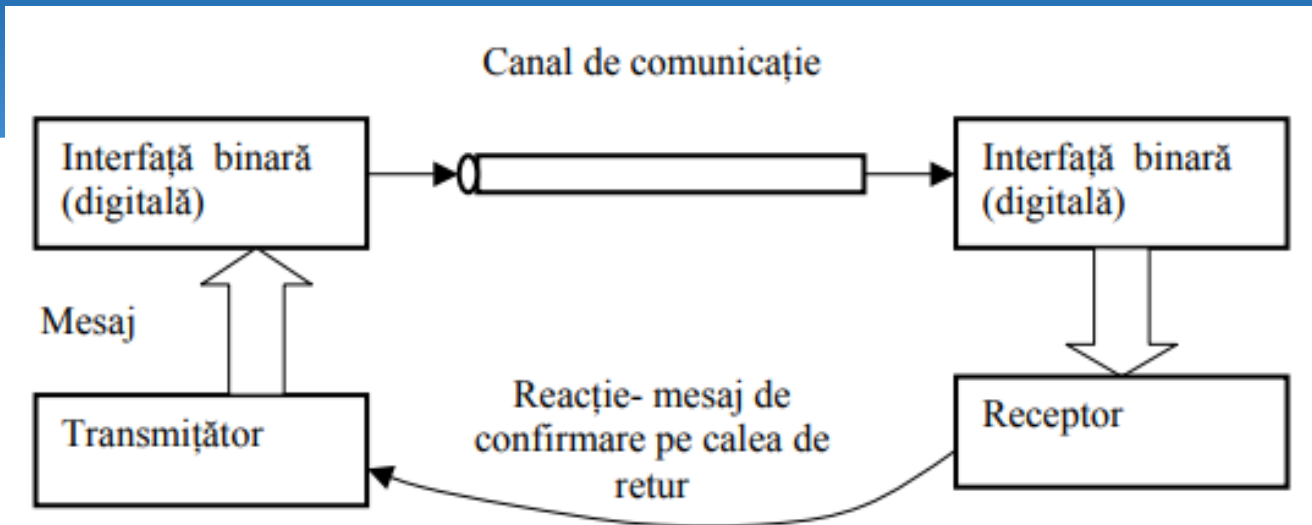
**Tema Nr. 6.1. Interfete si Protocoale de  
Comunicare.**

**Titular de curs:  
Conf.univ.,dr. V. Ababii**

# Subiecte abordate:

- Controlorul I8255. Structura. Principiul de funcționare. Programarea și metode de utilizare la organizarea schimbului de date în cod paralel.
- Controlorul I8251. Structura. Principiul de funcționare. Programarea și metode de utilizare la organizarea schimbului de date în cod secvențial.
- Interfața UART/USART
- Interfața COM.
- Magistrala USB.
- Interfețe RS: RS-232, RS-422, RS-423, RS-485.
- Interfața Bluetooth.
- Interfața IrDA.
- Interfața I2C.
- Interfața SPI.
- Interfața CAN.
- Interfața Ethernet.
- Controloare specializate pentru implementarea interfeței Ethernet. Protocoale de comunicare.

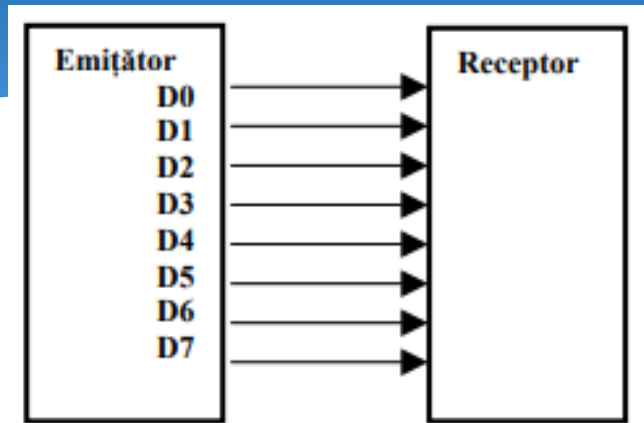
# Componentele procesului de comunicare.



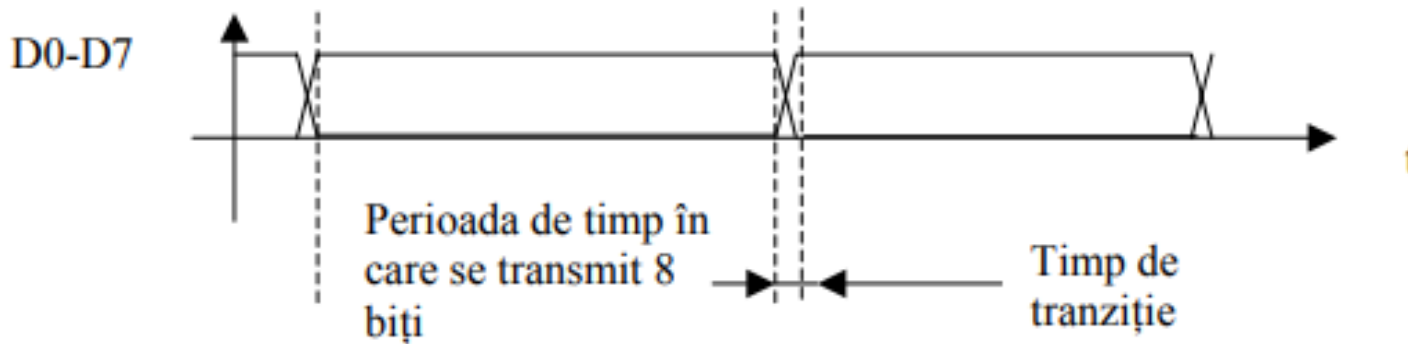
Un proces de comunicații de date necesită cel puțin 5 elemente,

- Transmițător;
- Mesaj;
- Interfață binară (digitală);
- Canal de comunicație;
- Receptor.

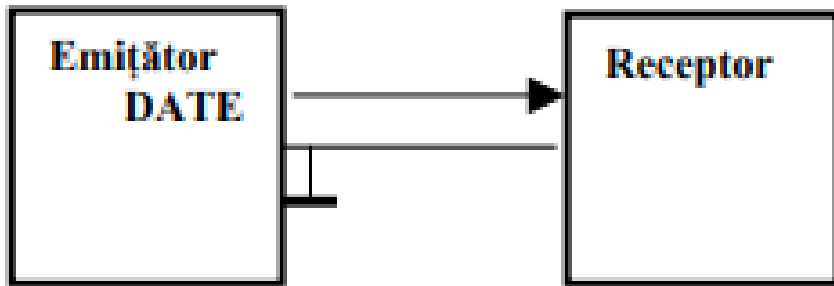
# Schema bloc a interfetei paralele de comunicare pe 8 biti.



Sunt utilizate pentru transferul de date la distanta mica, in interiorul PC.  
**Magistralele de Sistem:** ISA, EISA, PCI32/64, VLB, etc.  
**Interfata:** PRN, LPT, IDE, ATA, SCSI, etc.

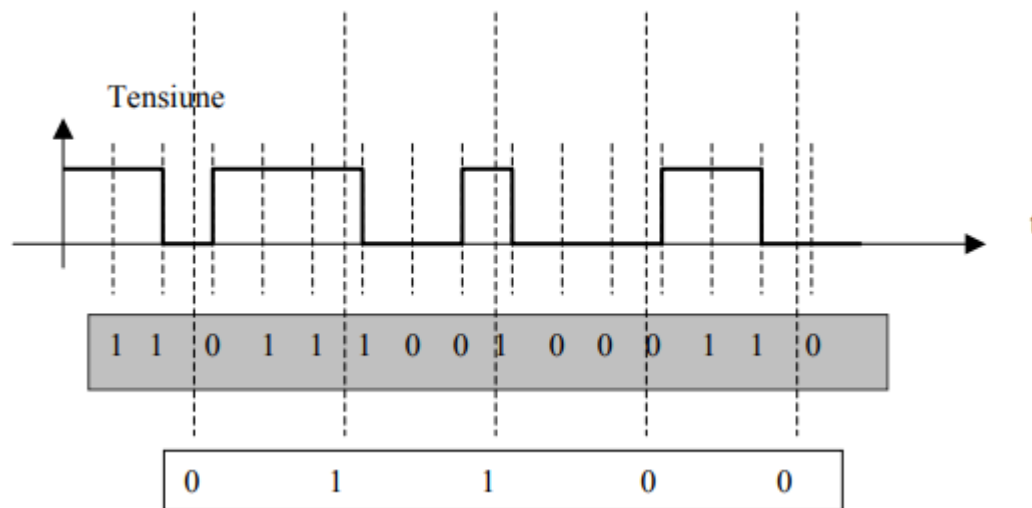


# Schema bloc a interfetei seriale de comunicare.



Sunt utilizate pentru transferul de date la distanța mare, schimb de date dintre PC.

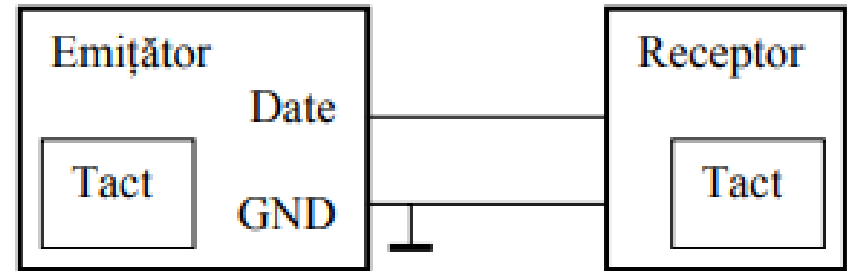
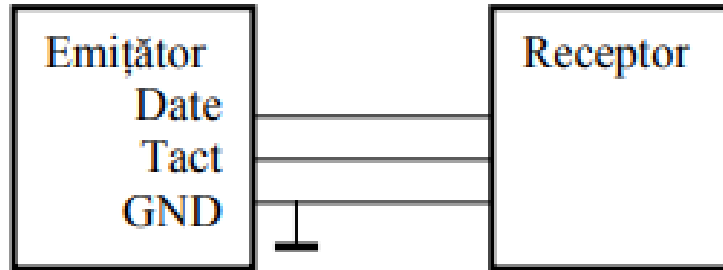
**Magistralele de Sistem:** Expres-PCI.  
**Interfata:** COM, UART, USART, I<sup>2</sup>C, SPI, CAN, RS, IrDA, USB, WiFi, etc.



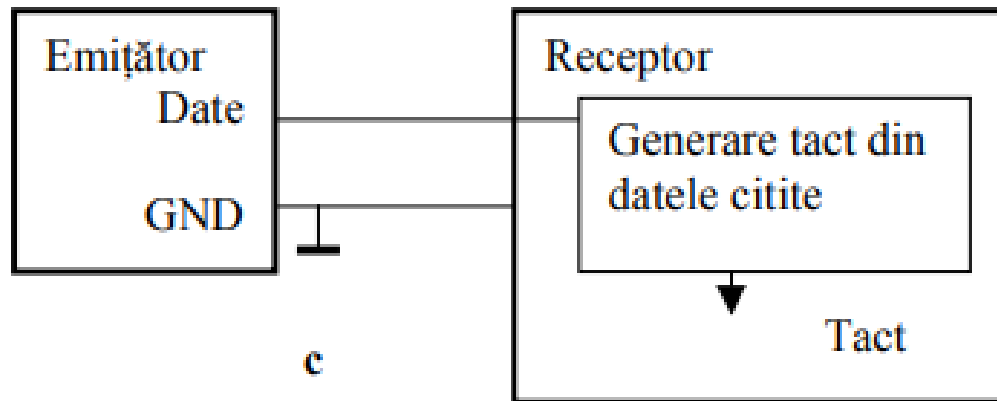
# Metode de sincronizare a transferului de date.

SPI

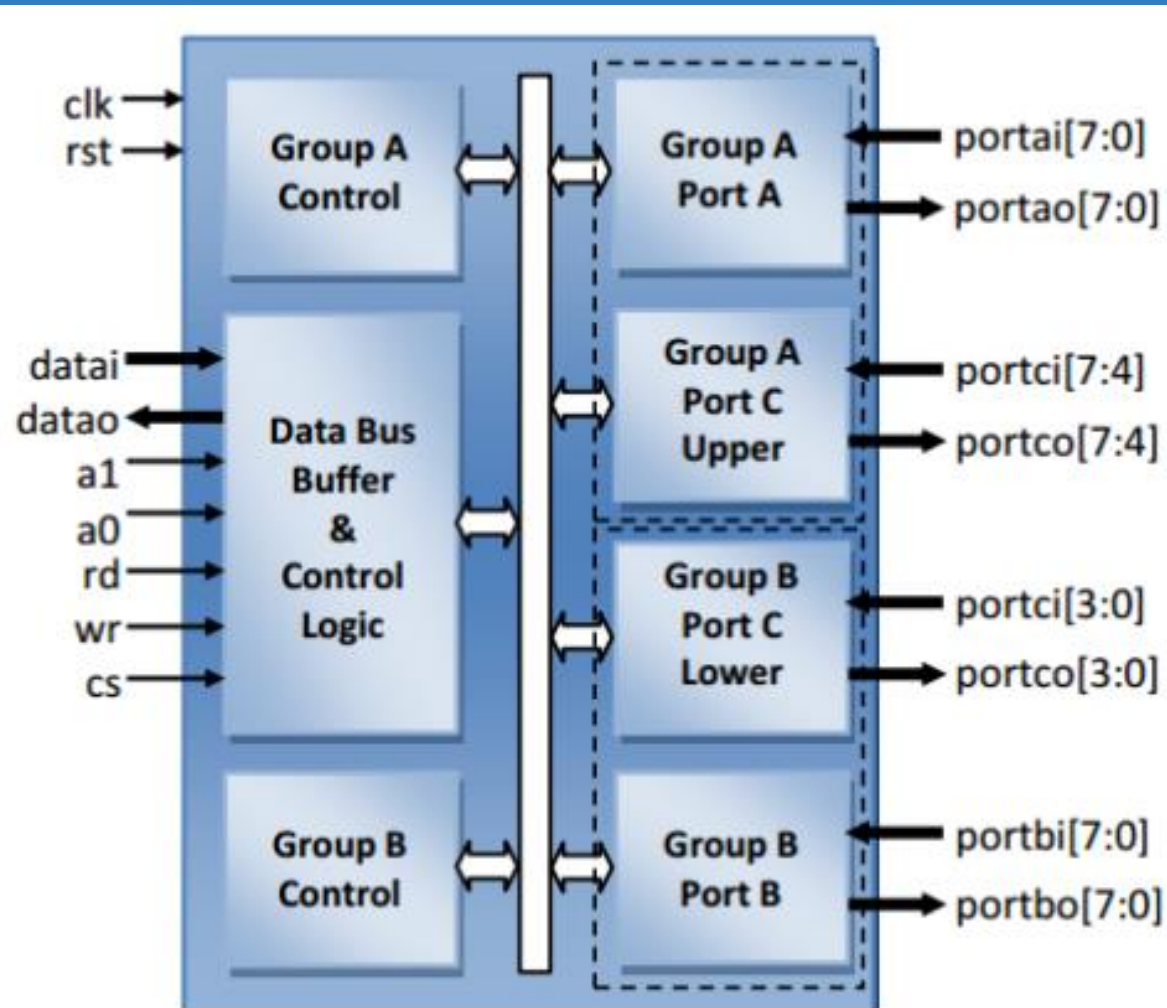
COM, RS, UART



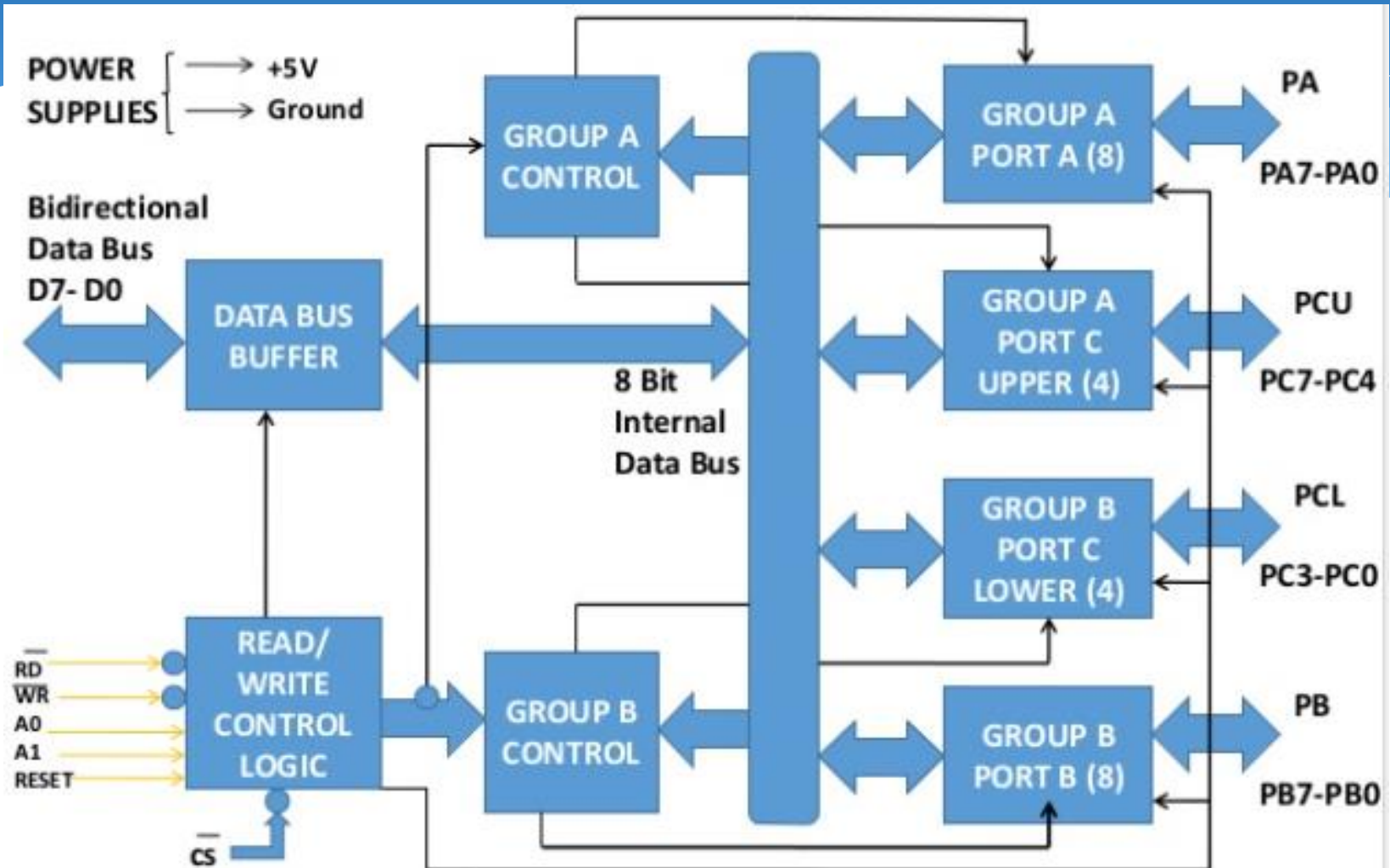
a USB, CAN, Ethernet, SATA, WiFi b



# Controlare pentru comunicare in cod paralel. Intel8255 (I8255)

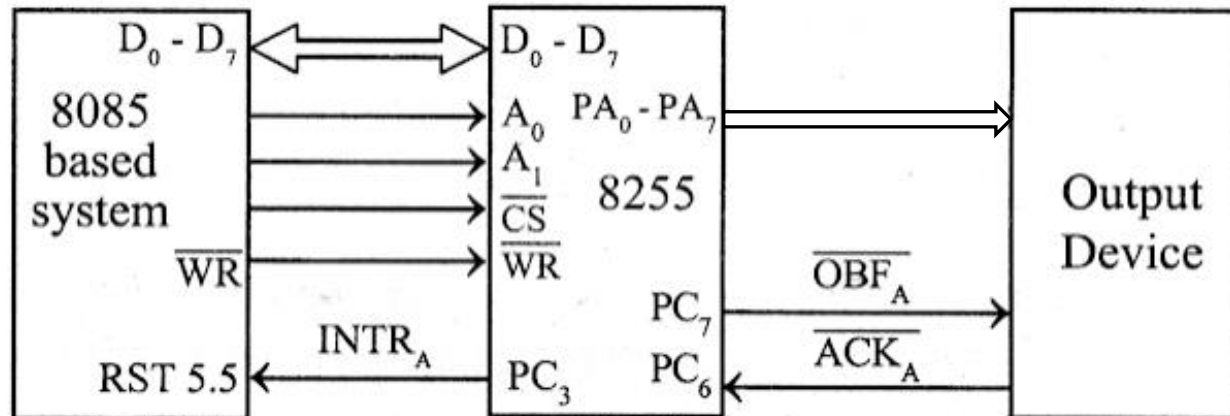
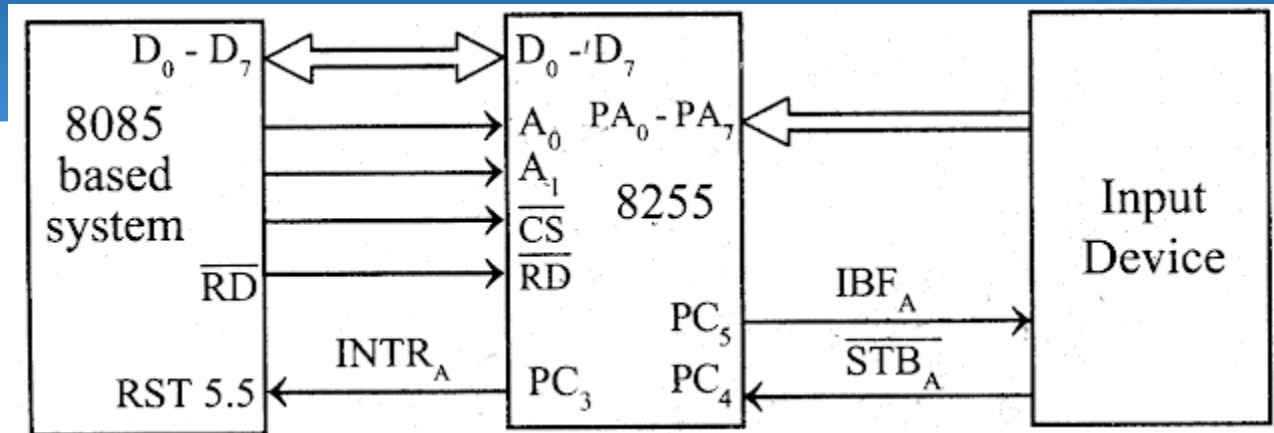


# Diagrama bloc detaliata I8255

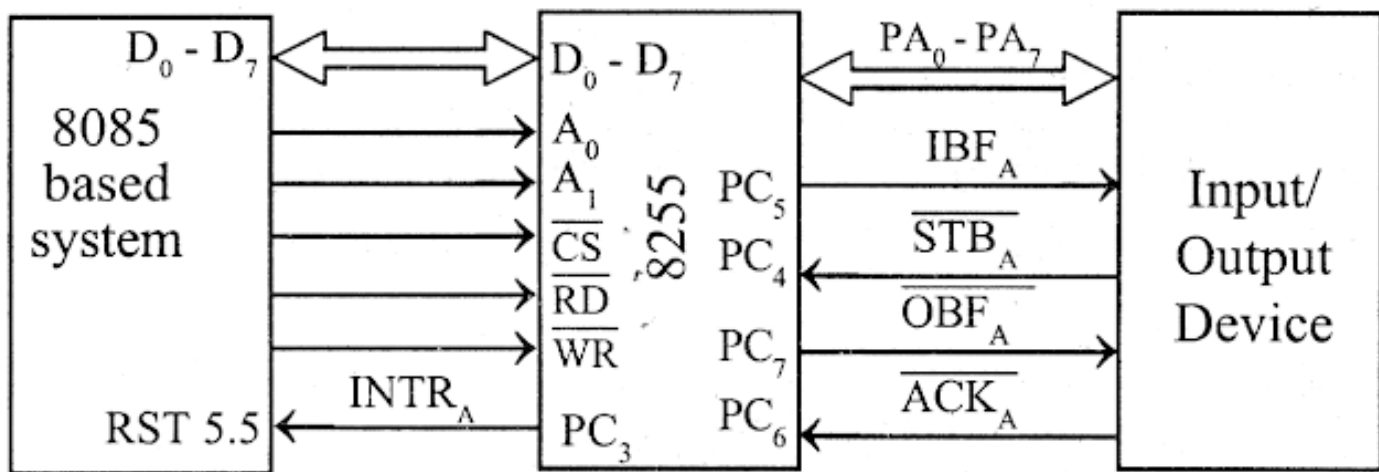




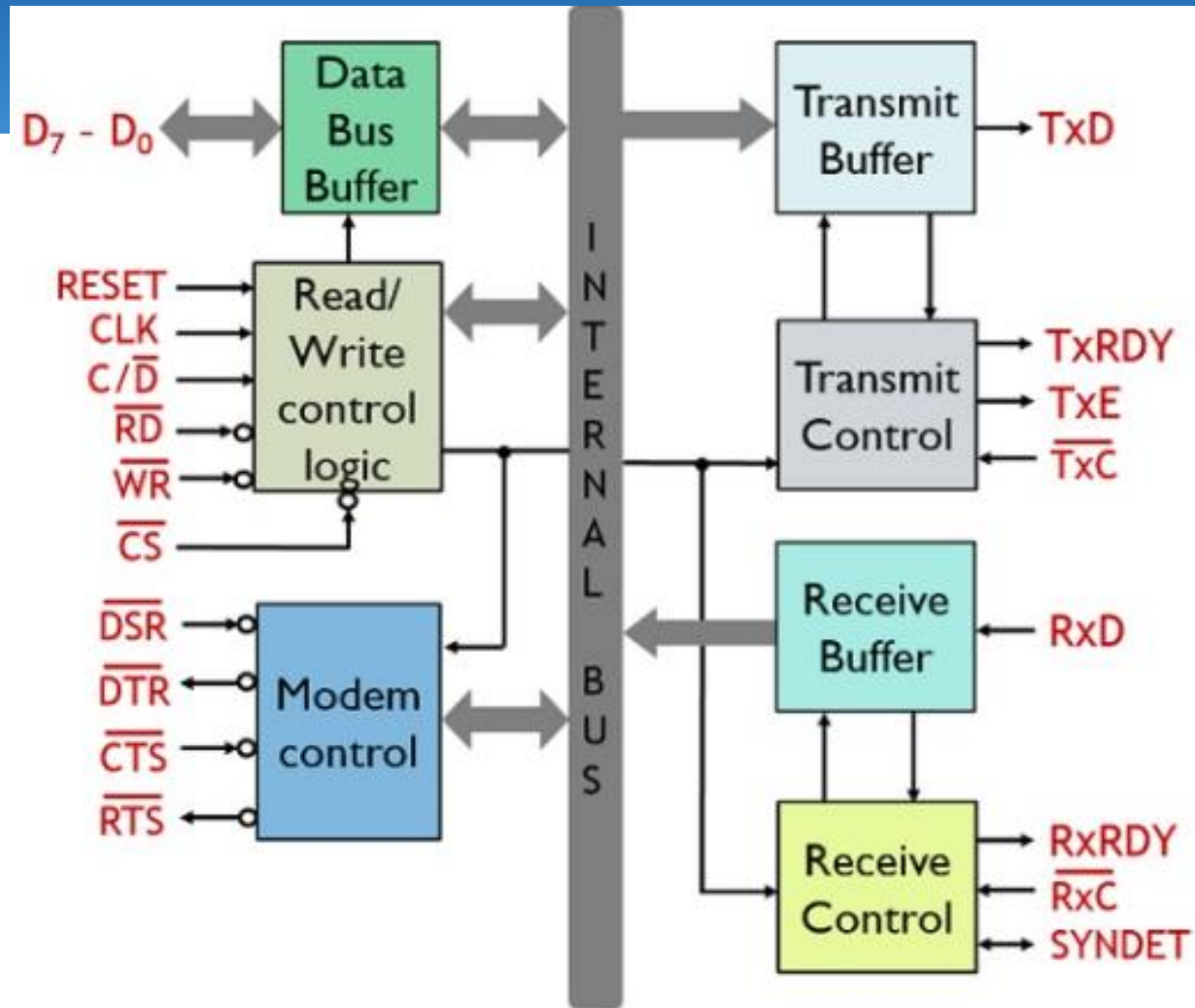
# Metode a aplicare I8255



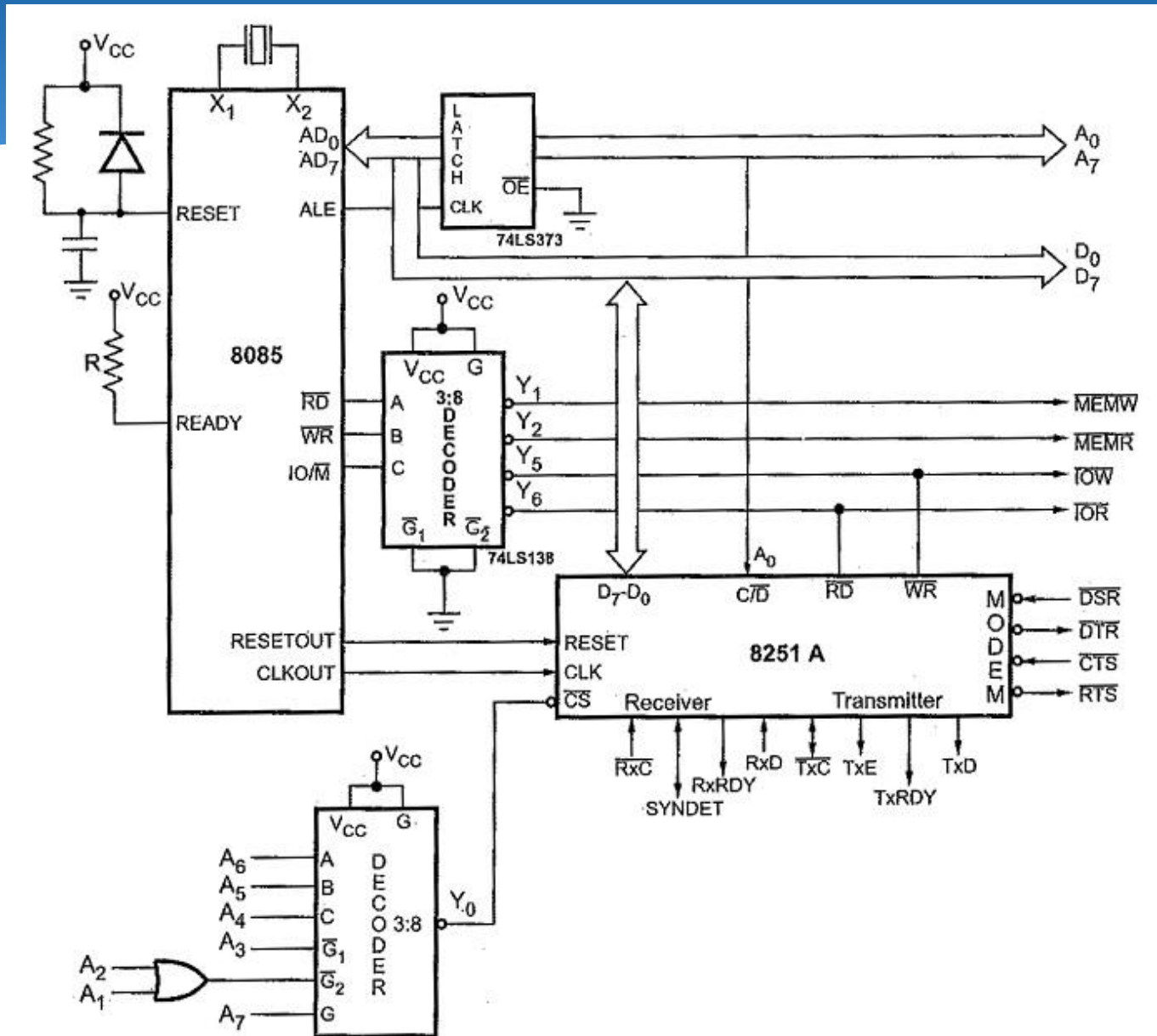
# Metode a aplicare I8255



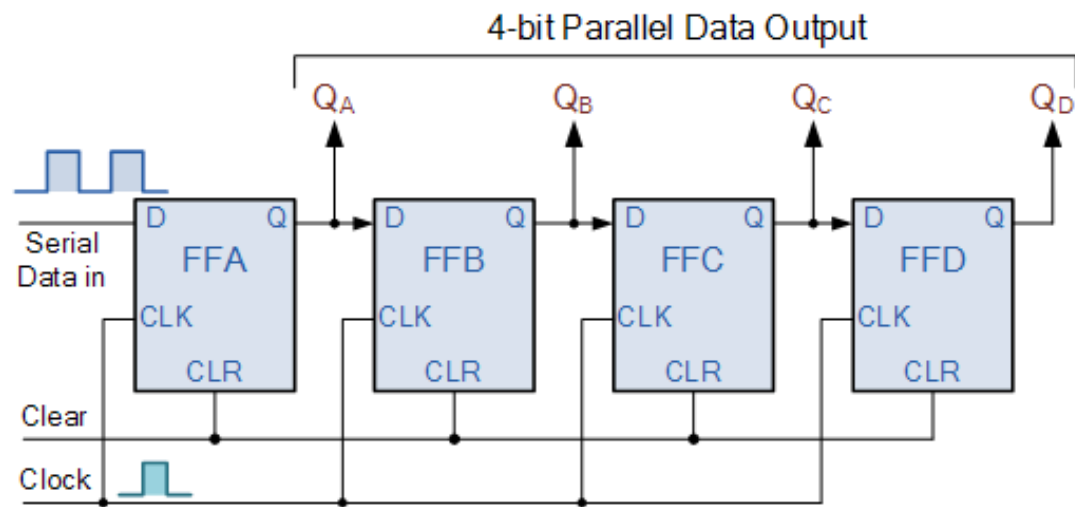
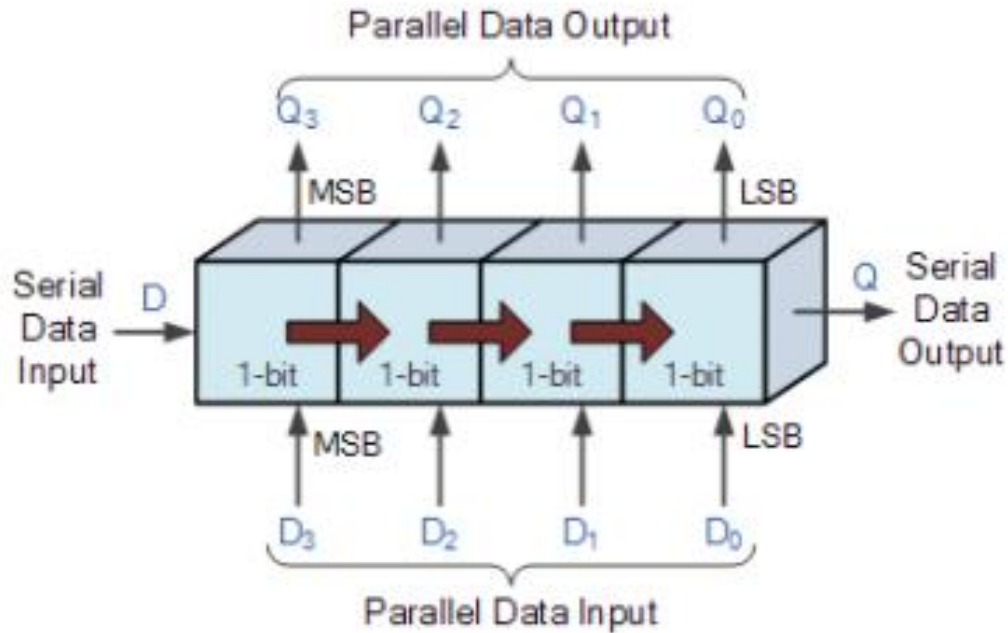
# Controlorul de comunicare in cod serie I8251



# I8251. Conectarea la MCU/MPU

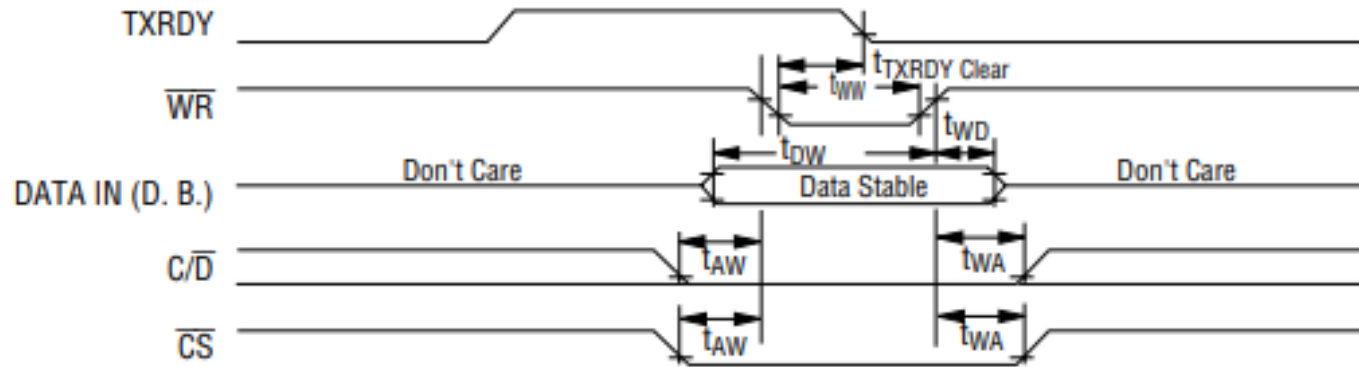


# 18251. Conversia datelor.

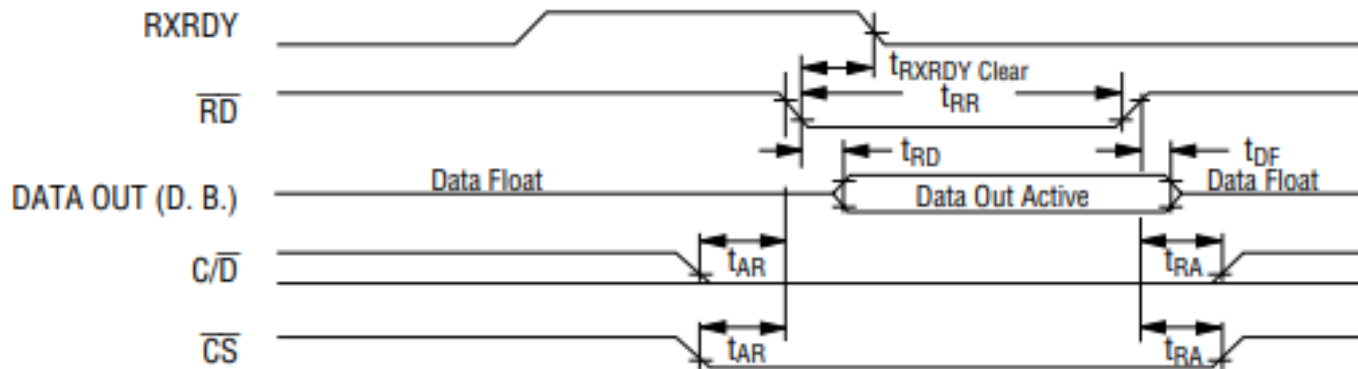


# I8251. Diagrama de timp.

## Write Data Cycle (CPU → USART)

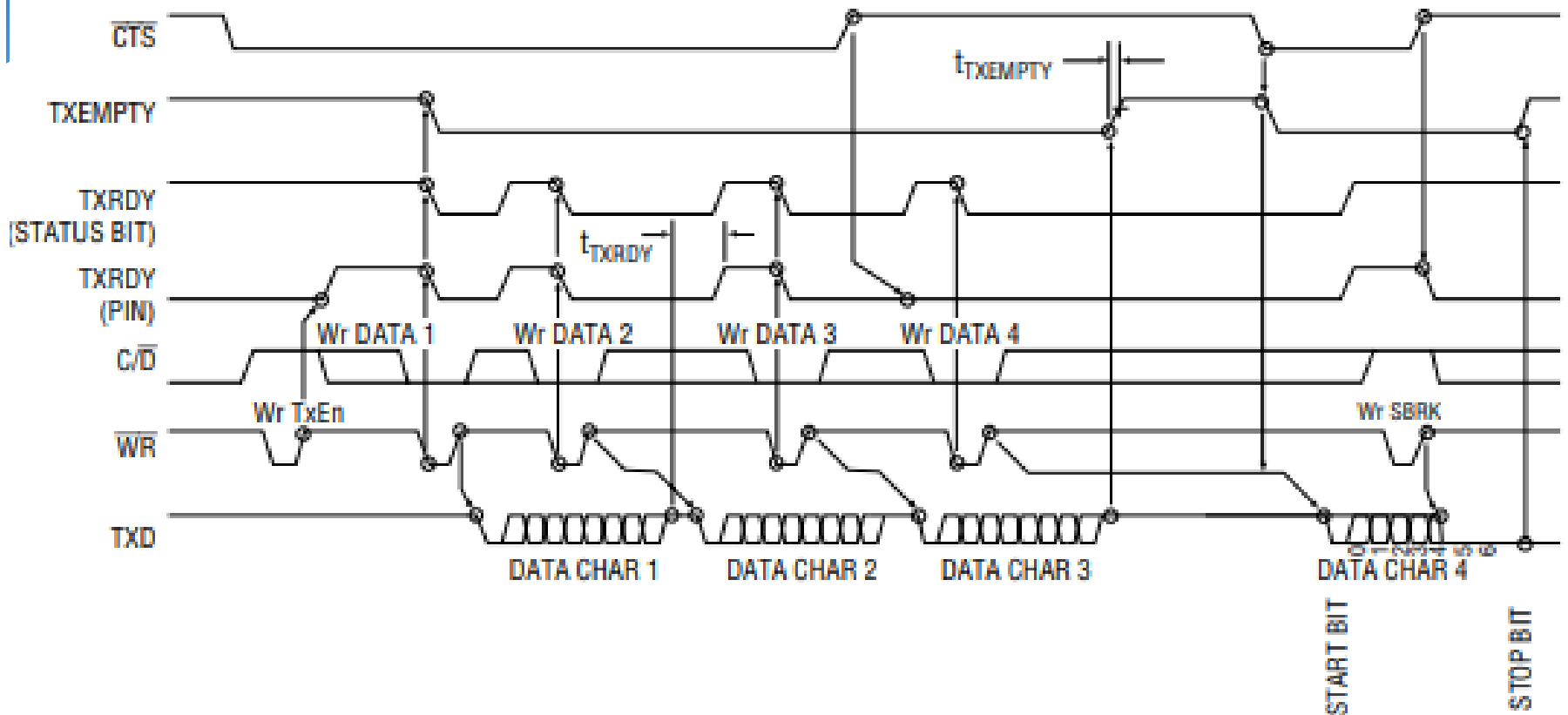


## Read Data Cycle (CPU ← USART)



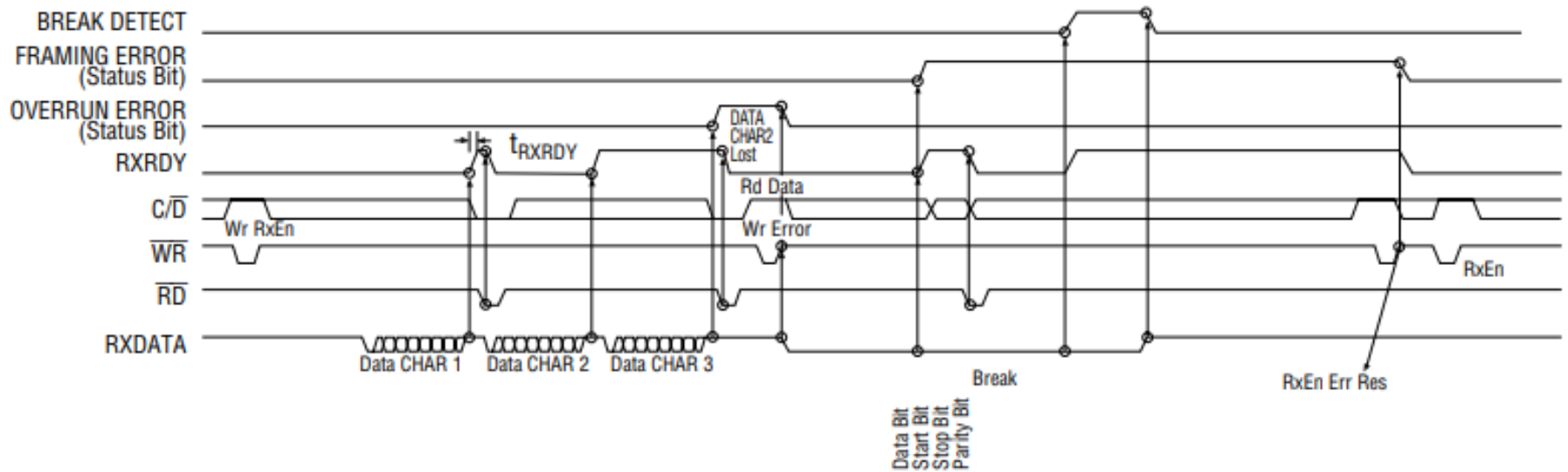
# I8251. Diagrama de timp.

## Regim TxD



# I8251. Diagrama de timp.

## Regim Rx D





# Interfata UART

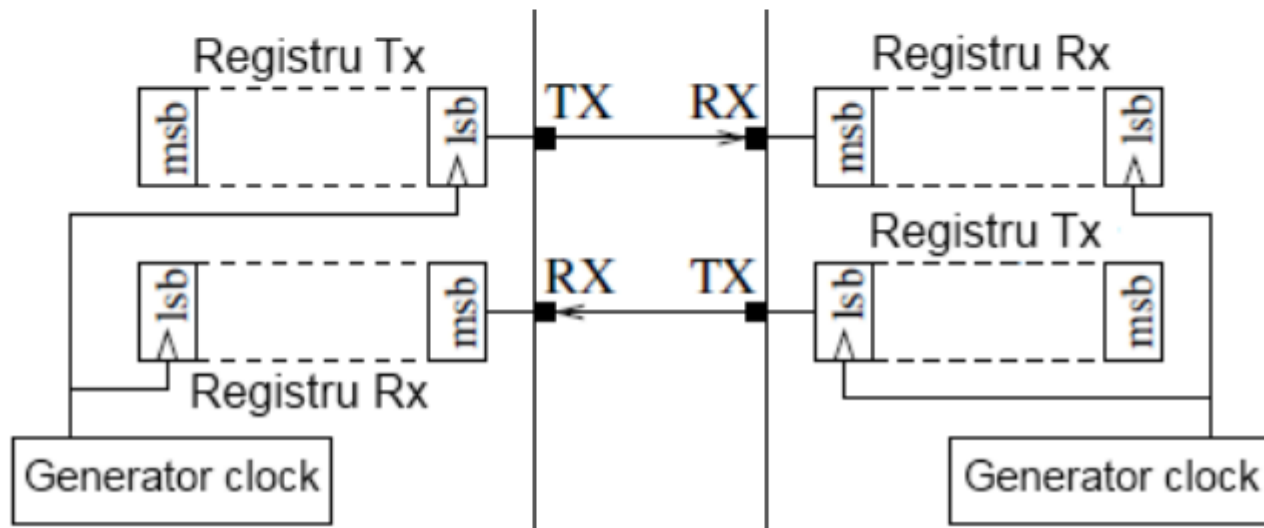
**Acronim: UART (*Universal Asynchronous Receiver Transmitter*)**

**Modul utilizat pentru comunicații asincrone**

**Utilizează două linii: receptor (RxD) și emitor (TxD)**

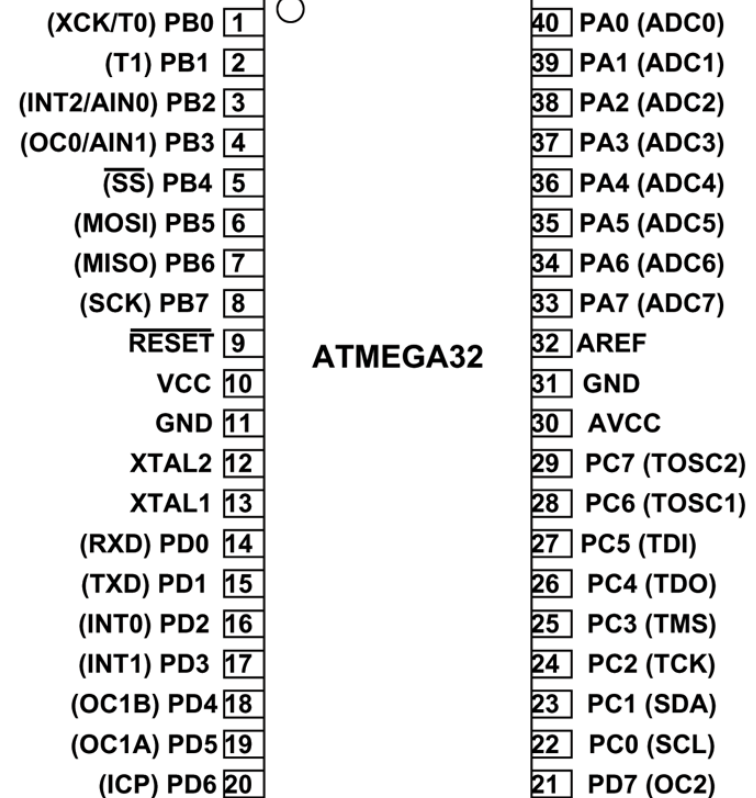
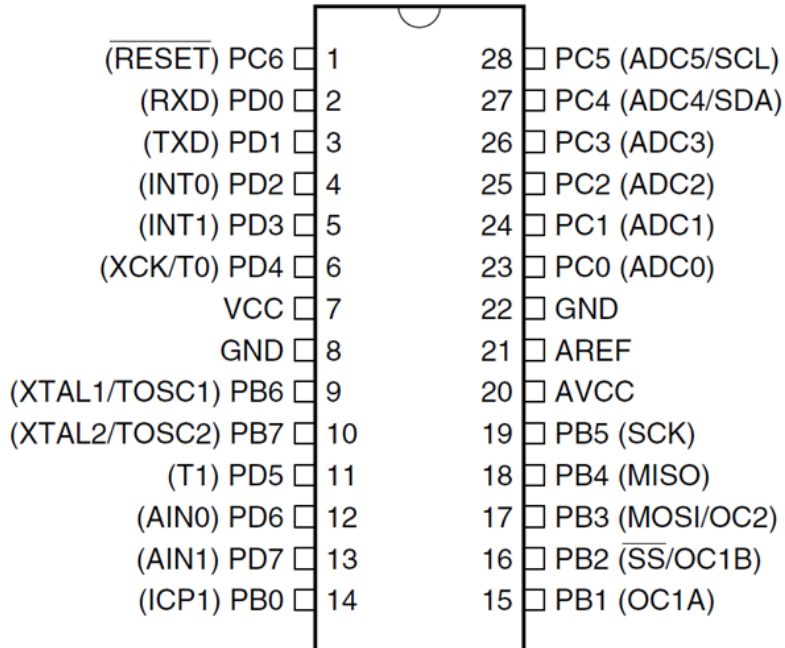
**Are mai mulți parametri configurabili:**

- **Număr biți date, Bit paritate, rata transfer, bit stop**

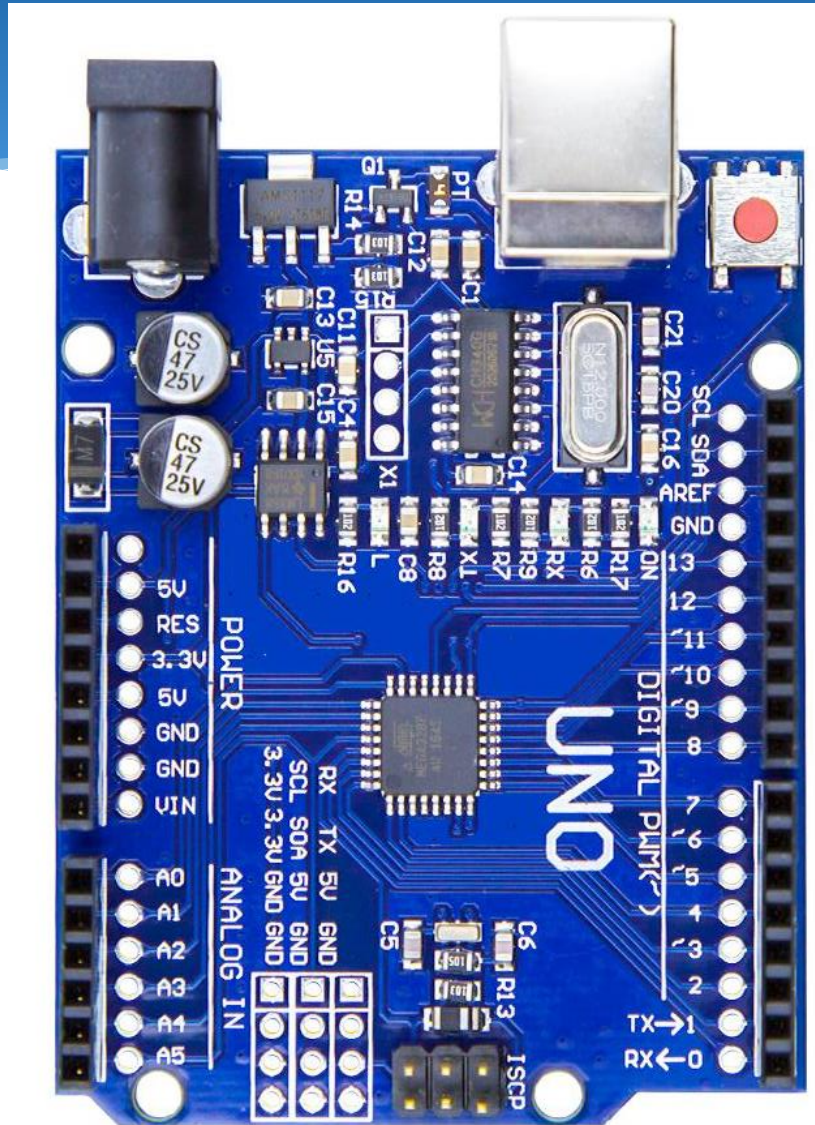


# Interfata UART. Domenii de aplicare.

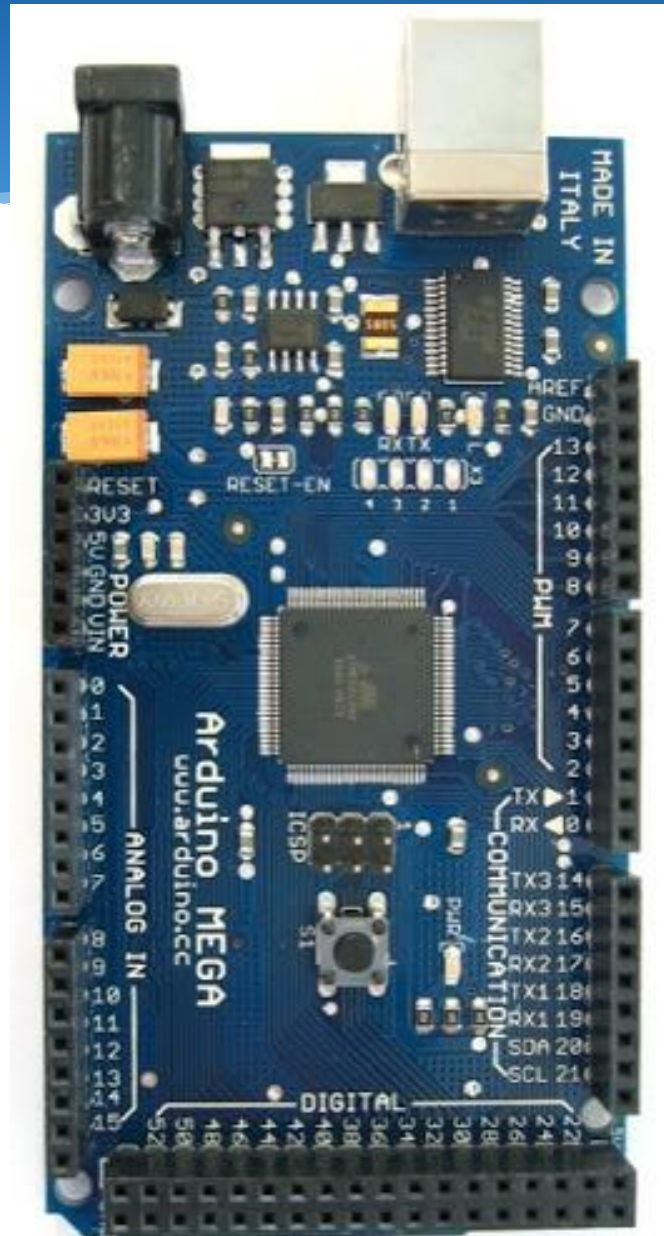
## ATMega8



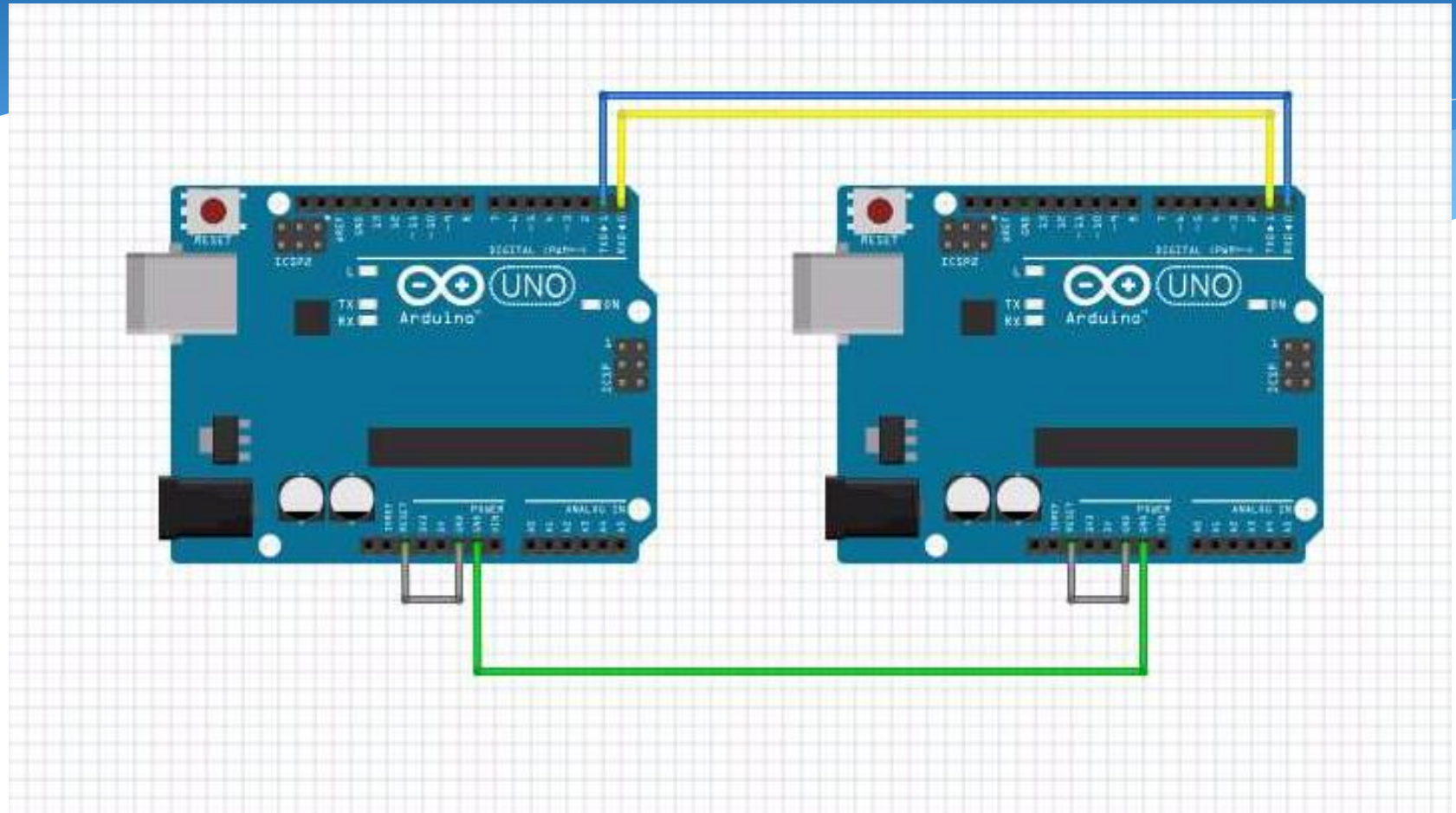
# Interfata UART. Domenii de aplicare.



# Interfata UART. Domenii de aplicare.



# Interfata UART. Domenii de aplicare.



# Interfata UART. Parametri de configurare.

## Număr biți date:

- **Dependent de producător, în general între 5 și 9 biți de date**

## Bit paritate

- **bit care să marcheze paritatea: pară sau impară**
- **Ex. paritate pară: bit paritate = 1 dacă nr. biți din mesaj este par**

## Rată transfer

- **Viteza de transmisie (biți/sec → bps)**
- **Valori cuprinse între 9600-115200 (depinde de clock)**

## Bit stop

- **Permite utilizarea unui singur bit de stop, sau doi biți de stop**

# Interfata UART. Parametri de configurare.

Formatul care permite descrierea configurației UART este:

**$D\{E|O|N\}S$**

- **D** – numărul de biți de date
- **S** – numărul de biți de stop
- **E|O|N** – paritate pară, impară sau fără paritate
- **Ex: 8biți date, paritate pară, 1 bit stop:**

**$8E1$**

# Interfata UART. Codificarea fizica a datelor.

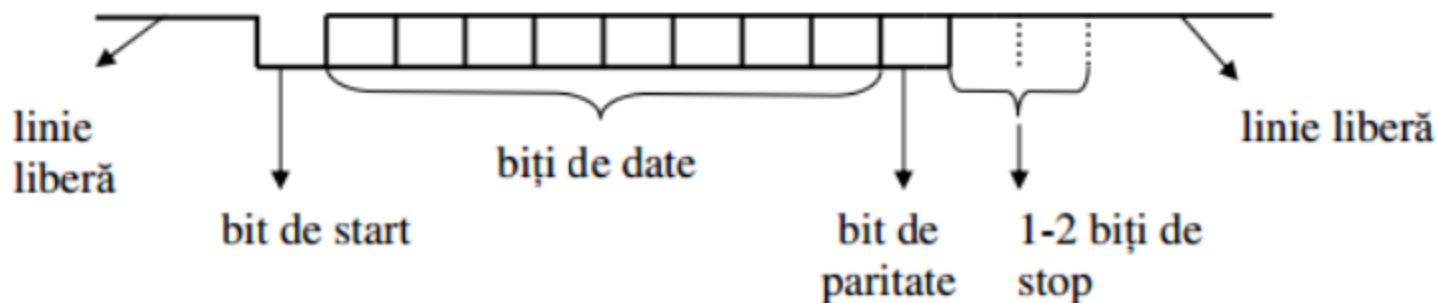
Utilizează NRZ (*Non Return to Zero*)

Următoarea structură de date este transmisă:

- un bit de start (0 logic)
- 5-8 biți de date
- 0-1 bit de paritate (pentru detecția erorilor)
- 1-2 biți de stop (1 logic)

În starea de așteptare (liberă) linia este High

Cel mai puțin semnificativ bit este transmis primul





# Interfata UART. Gestiunea erorilor.

Se utilizează bitul de paritate pentru depistarea erorii (1 bit eroare)

- Bitul de *eroare\_paritate* din registrul UART este setat

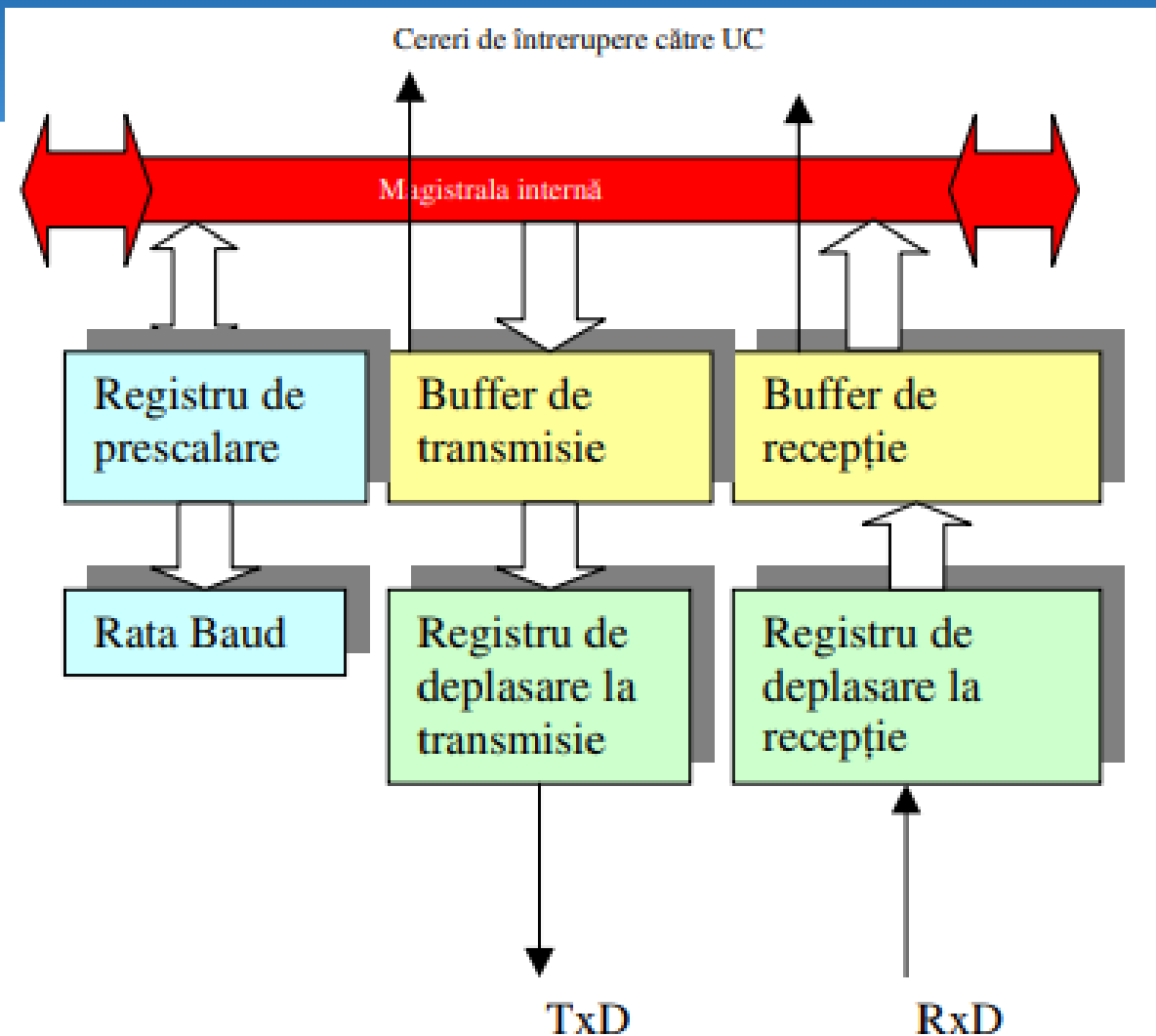
Divergența ratei de transfer dinte E și R conduce la pierderea sincronizării

- Bitul de stop nu este recunoscut cu nivel High
- Bitul de *frame\_error* din registrul UART este setat

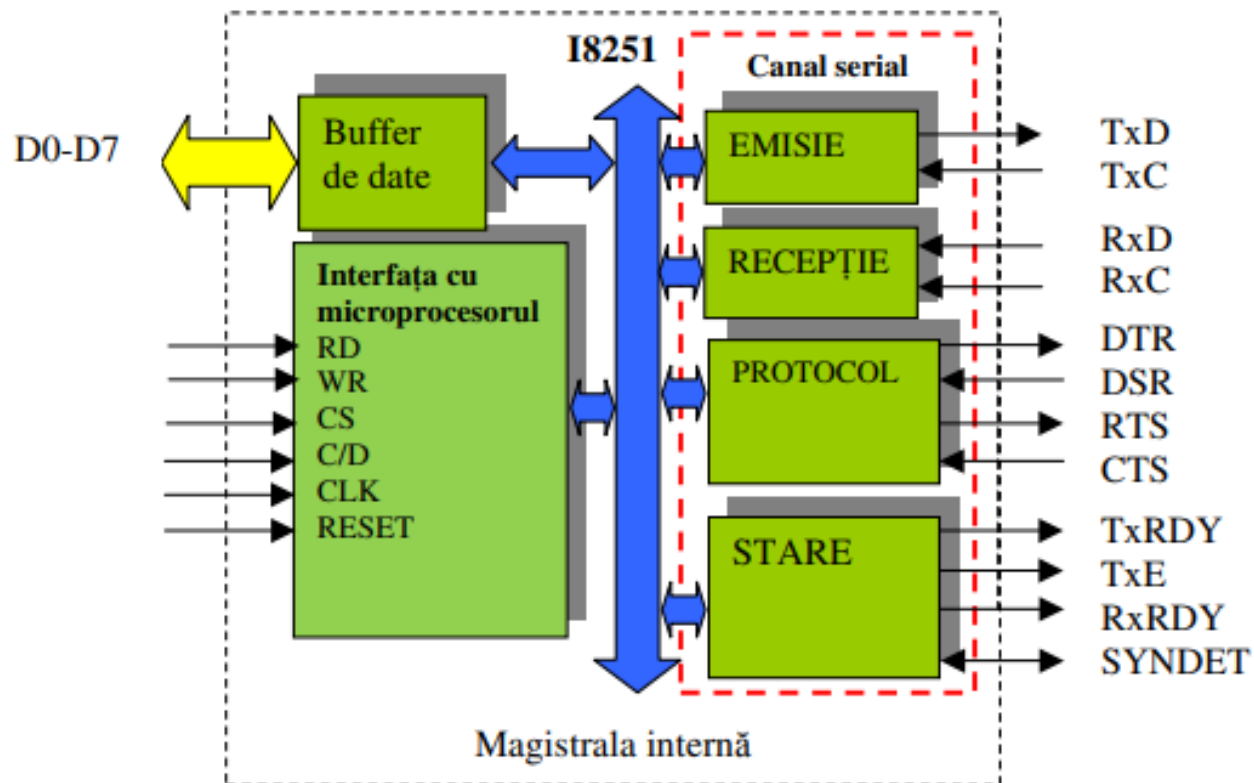
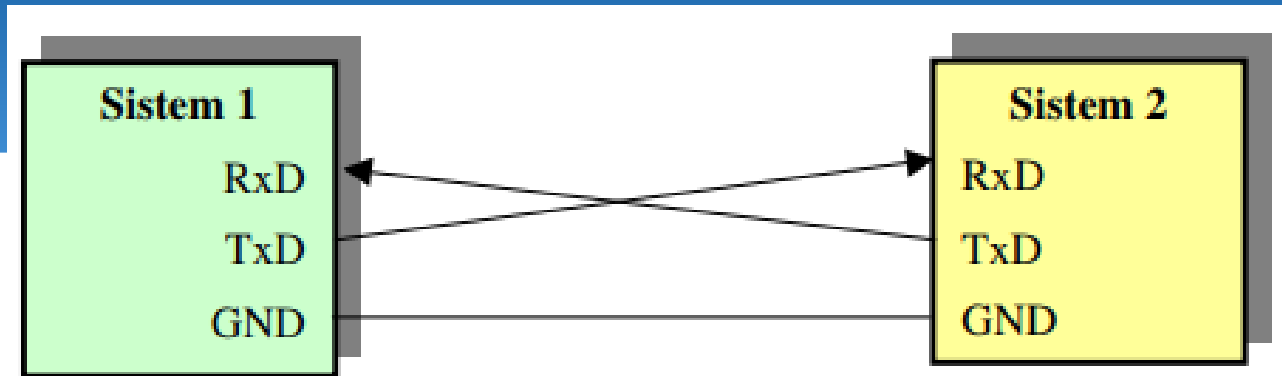
Datele din registrul buffer nu au fost citite în totalitate (o nouă transmisie a început)

- Datele din registrul de deplasare sunt pierdute
- Bitul de *data\_overrun* din registrul UART este setat

# Interfata UART. Structura Interna.



# Interfata UART. Interconectarea dispozitivelor.



# Interfata UART. Programarea in Arduino IDE.

```
int incomingByte = 0; // for incoming serial data
```

```
void setup() {  
  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps  
}
```

```
void loop() {  
  // receive data:  
  if (Serial.available() > 0) {  
    // read the incoming byte:  
    incomingByte = Serial.read();  
    Serial.write(incomingByte);  
    Serial.print("I received: ");  
    Serial.println(incomingByte, DEC);  
  }  
}
```

# Parametrii de comunicare seriala.

*Viteza de comunicație* (numită și *debit binar*) este măsurată în biți/s (bps):

$$D = \frac{1}{T} \text{ [biți/s]}$$

unde  $T$  este perioada de timp necesară pentru transmisia sau recepția unui singur bit.

Modemul reprezintă semnalele de date prin diferite stări electrice, în funcție de tipul de modulație pe care îl utilizează: frecvență, amplitudine, sau fază. Fiecare stare electrică este menținută la ieșirea modemului pentru un interval de timp numit *perioadă de modulație* ( $\Delta$ ). *Viteza de modulație* este inversul perioadei de modulație, reprezentând numărul schimbărilor pe secundă ale stării electrice a modemului:

$$V_m = \frac{1}{\Delta} \text{ [baud]}$$

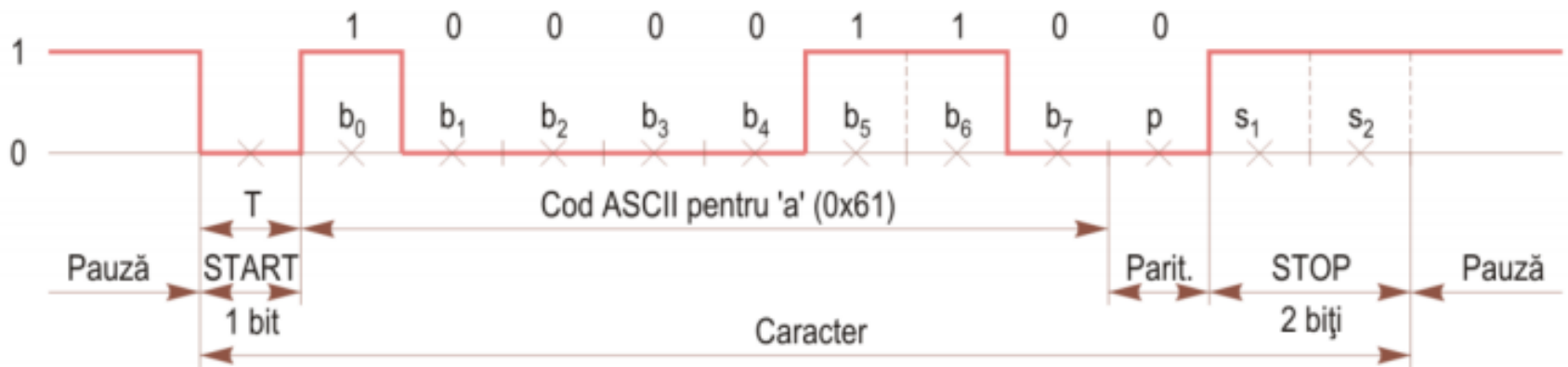
Unitatea de măsură a vitezei de modulație este *baud*, după numele inginerului și telegrafistului francez Jean-Maurice Baudot. Relația dintre viteza de comunicație  $D$  și viteza de modulație  $V_m$  este:

$$D = V_m \cdot \log_2 n \text{ [biți/s]}$$

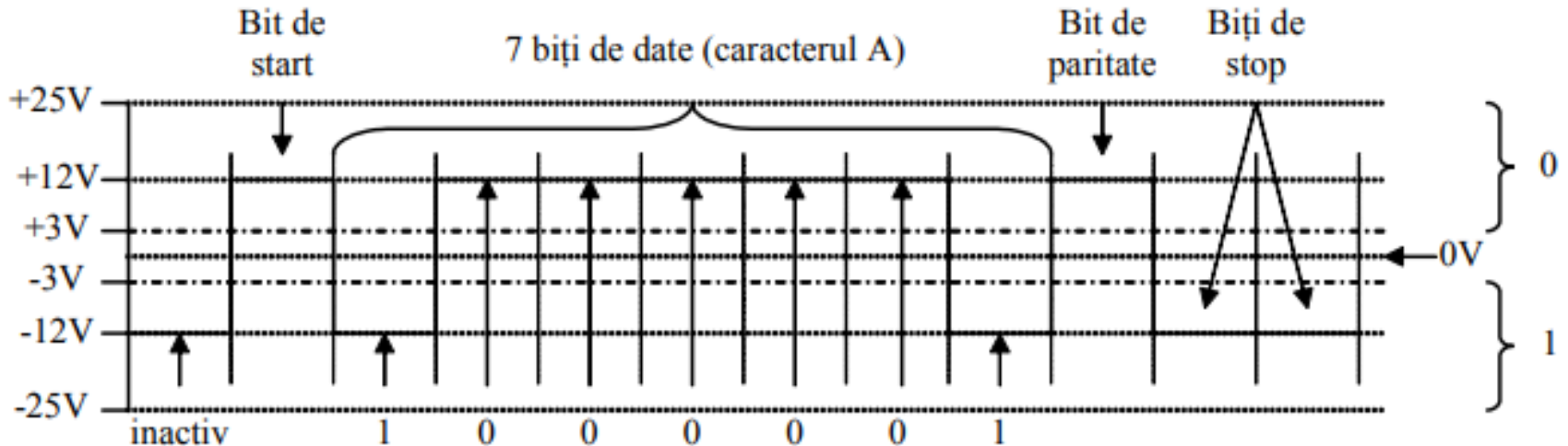
unde  $n$  este numărul stărilor electrice distincte ale modemului. În cazul particular când există doar două stări electrice distincte ale modemului, viteza de comunicație este egală cu viteza de modulație. În general însă, există un număr mai mare de stări electrice ale modemului, astfel încât viteza de comunicație este un multiplu al vitezei de modulație.

# Interfata COM/RS-232.

## Formatul Datelor in UART

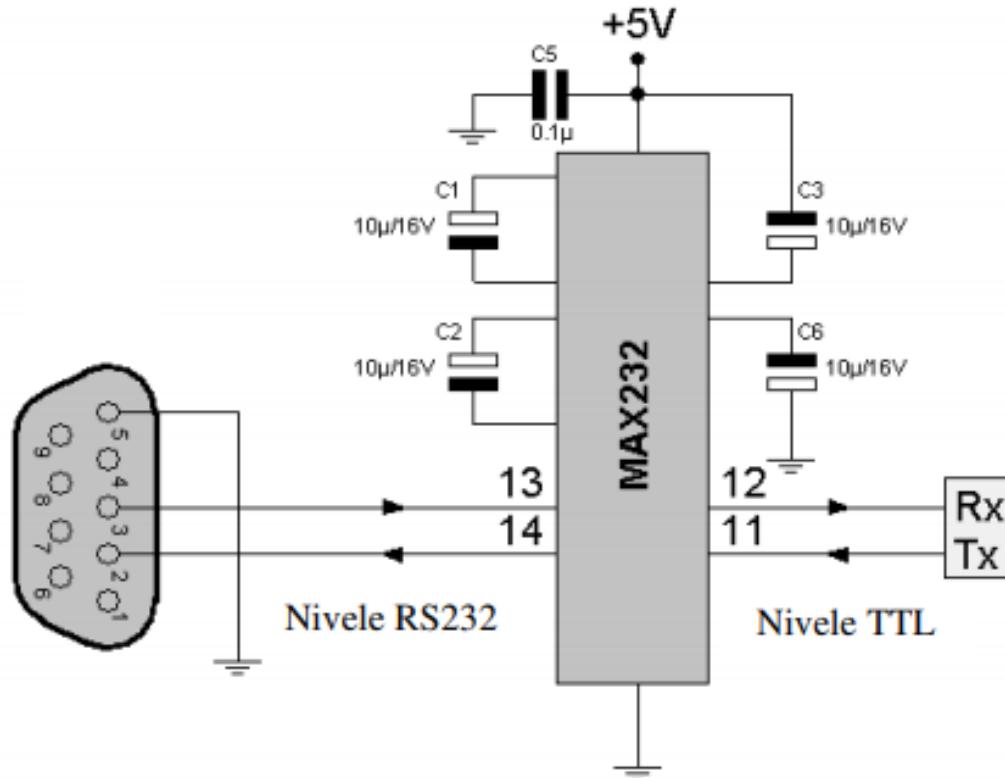


## Formatul Datelor in COM/RS-232



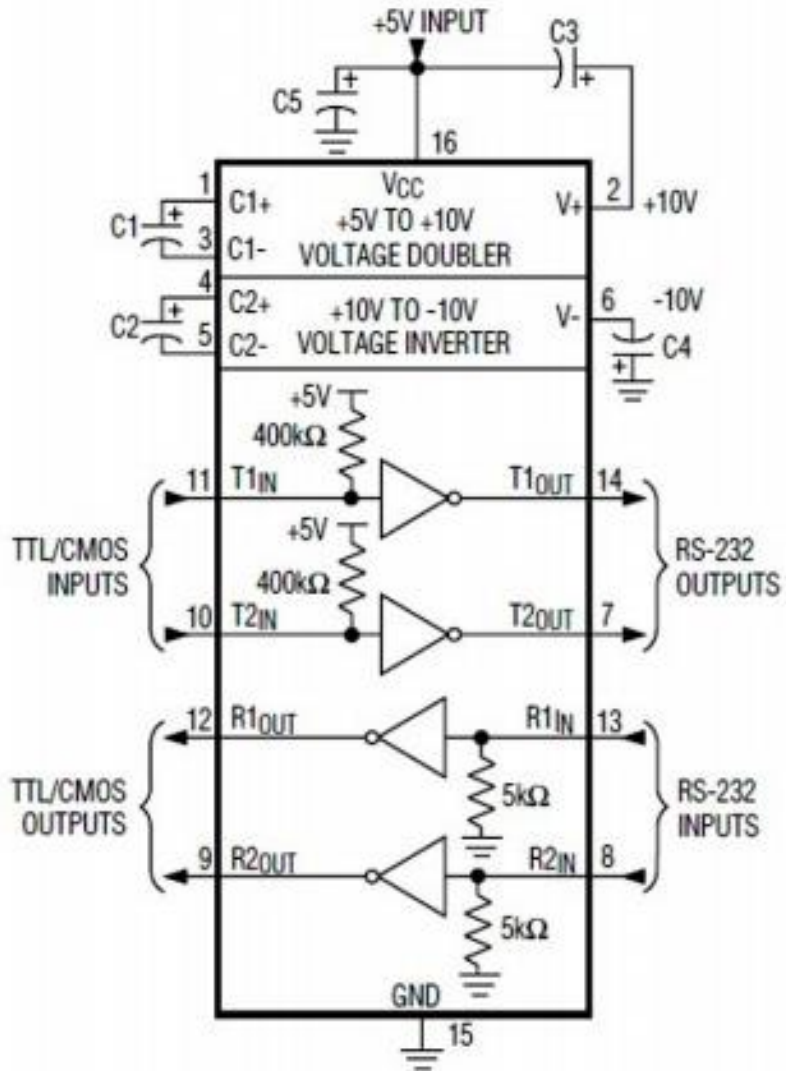
# Interfata COM/RS-232.

## Schema de conversie TTL – COM/RS-232



# Interfata COM/RS-232.

## Schema CI MAX232



MCU

COM



# Interfata UART/RS-232.

## UART si RS232:

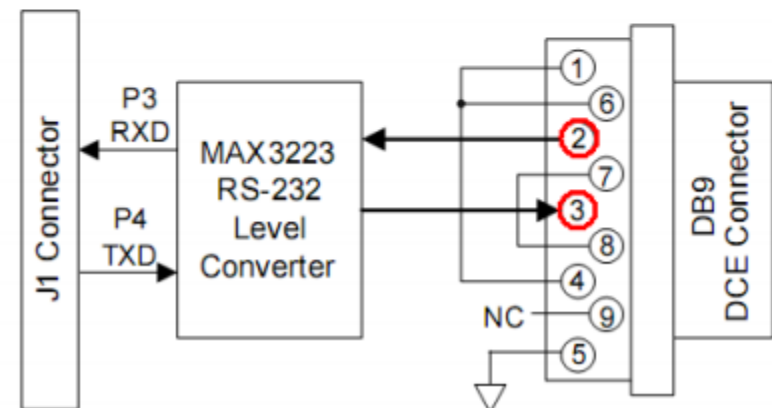
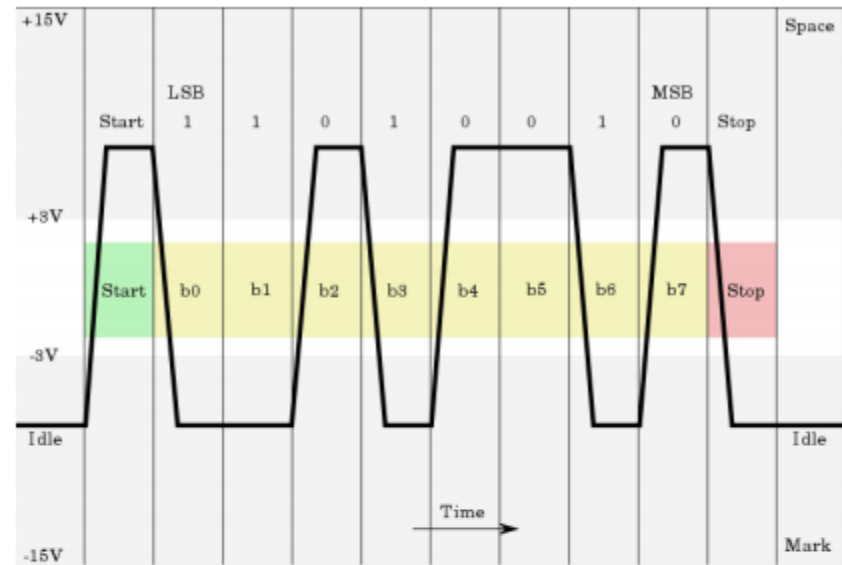
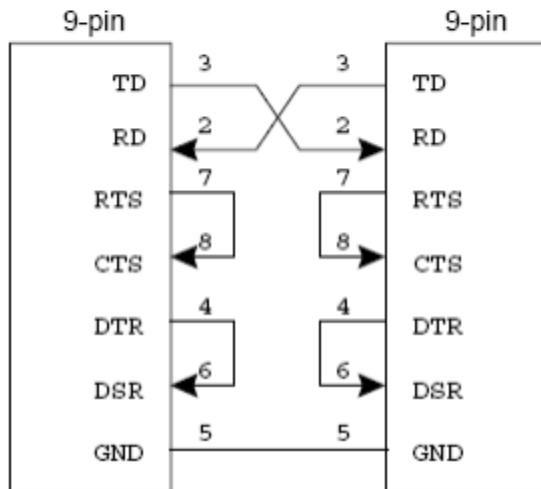
Adaptarea nivelelor de tensiune

RS232 logic '1' -5... -15 V

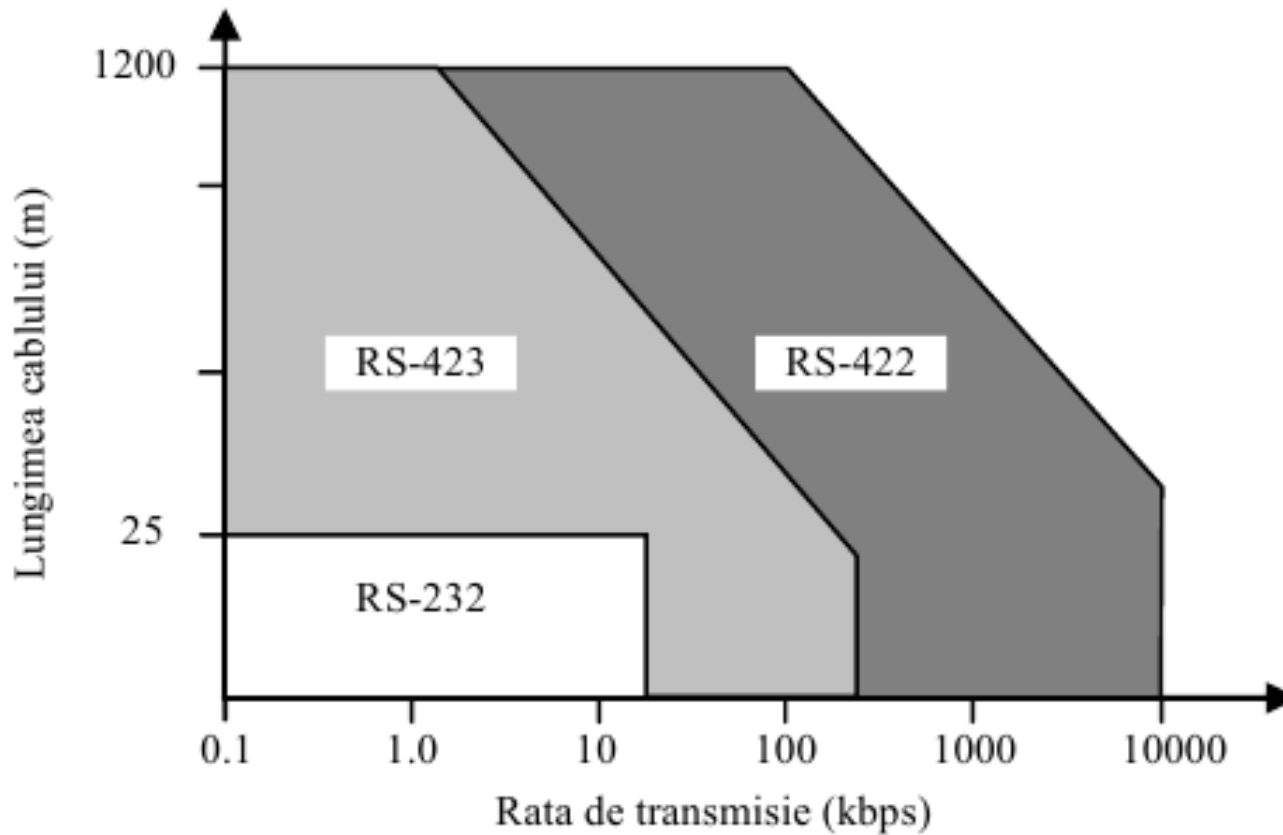
RS232 logic '0' +5...+15 V

Este nevoie de conversie de la nivelele logice UART la RS232

Corespondenta pinilor

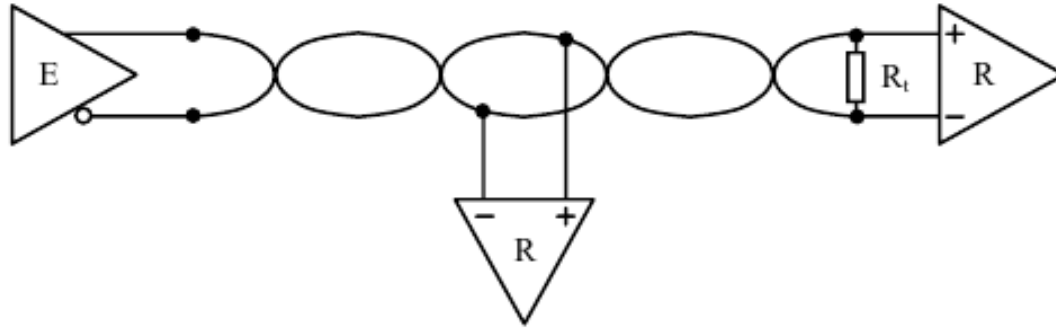


# Interfata RS-422.

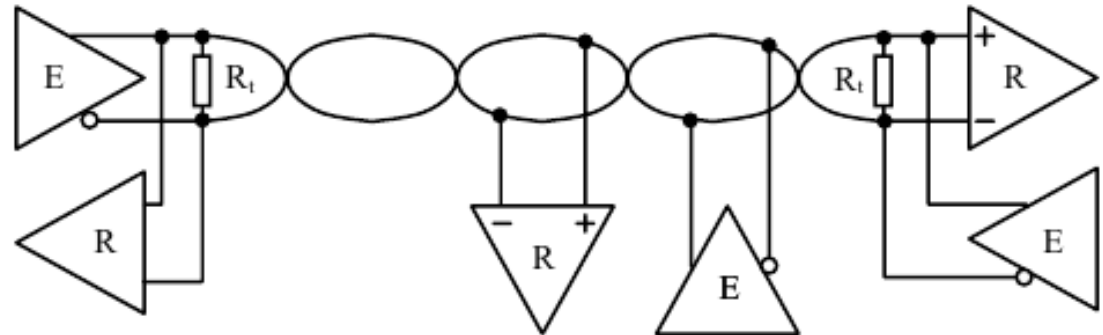


# Interfata RS-422.

Rețelele RS-422 folosesc 3 tipuri de configurații:

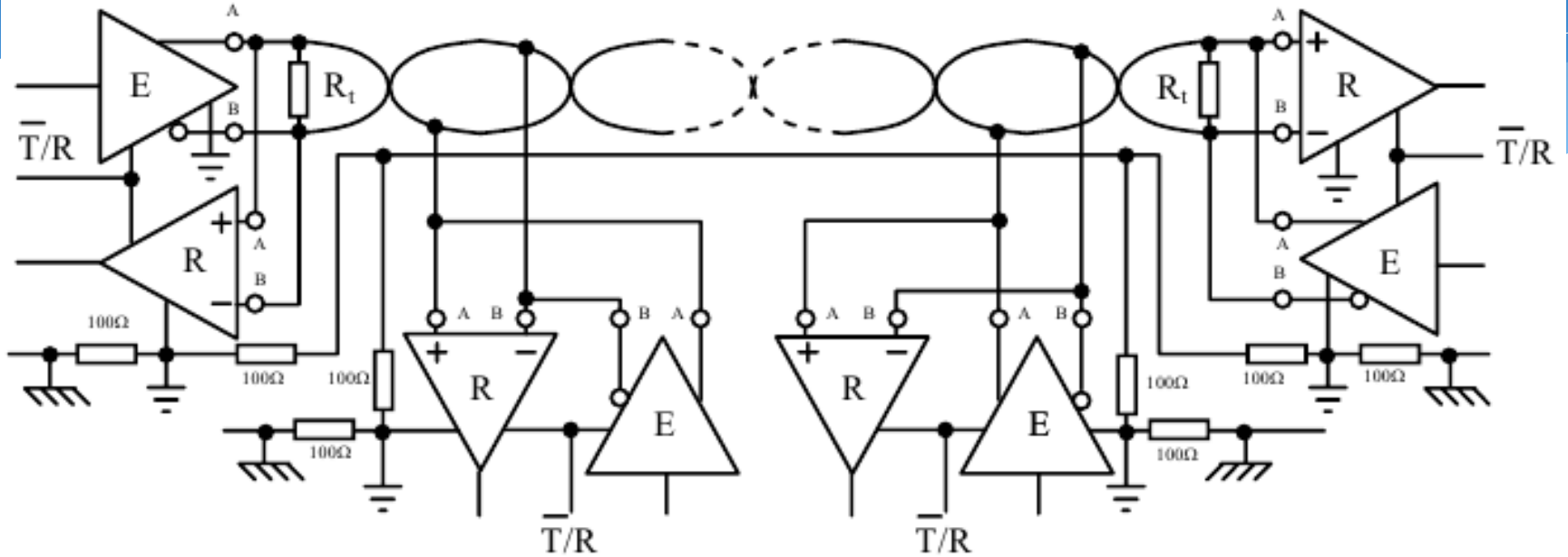


Aplicație multi-recepție cu terminator paralel



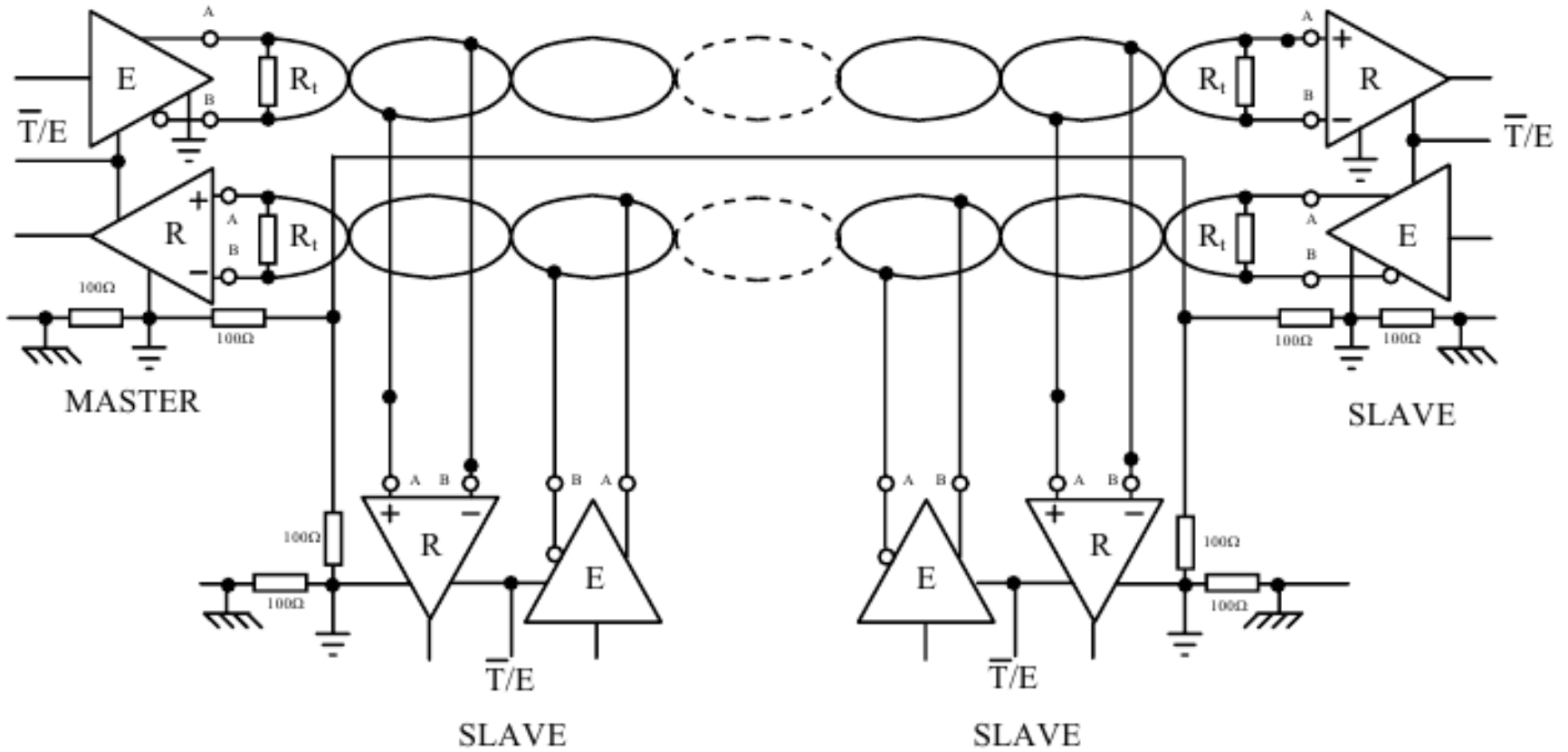
Aplicație multipunct-multipunct

# Interfata RS-485.



Aplicație RS-485 multipunct-multipunct pe 2 fire

# Interfata RS-485.



Aplicație RS-485 multipunct-multipunct pe 4 fire

# Interfata Bluetooth.