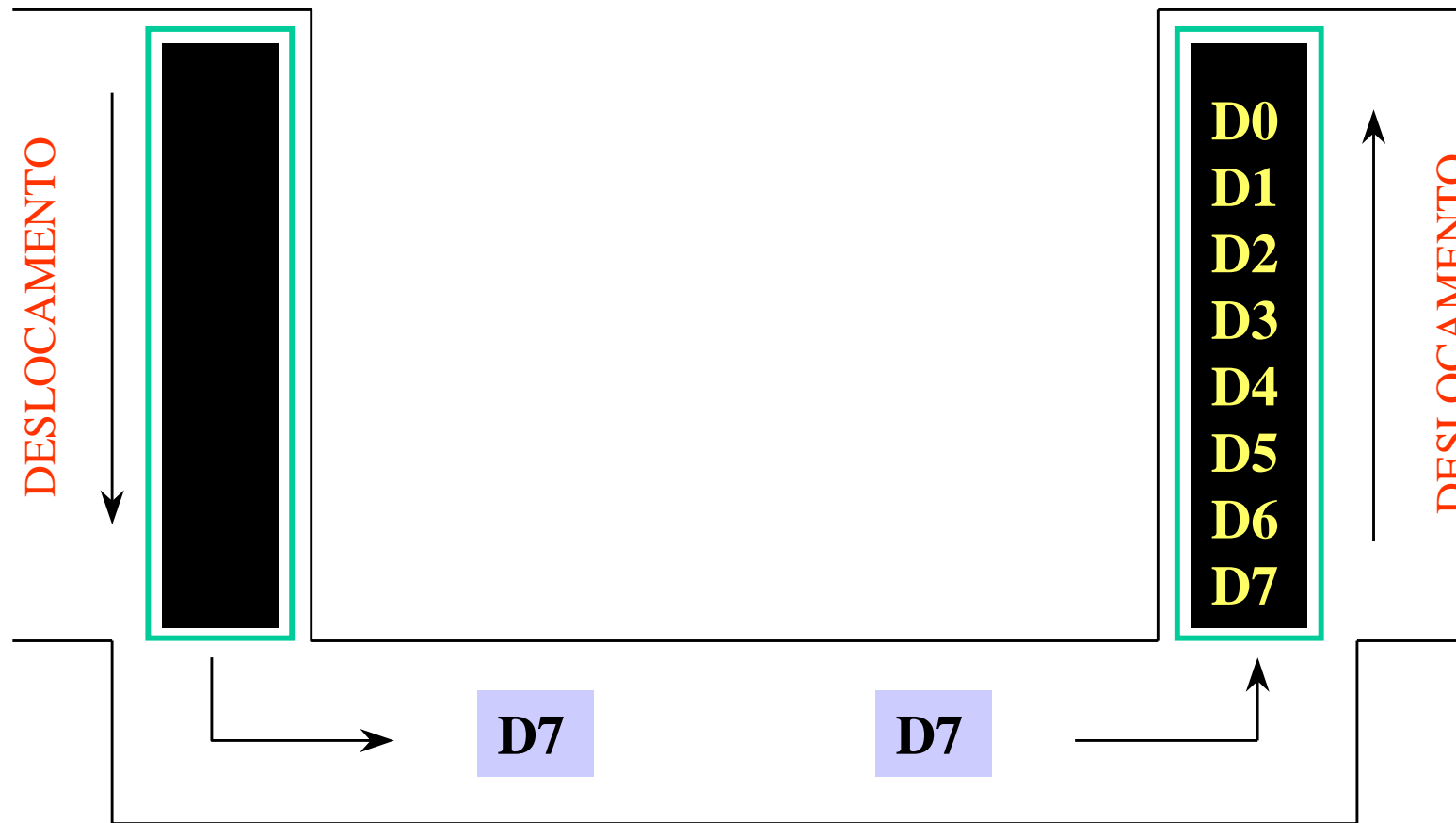


Comunicação Serial

TRANSMISSÃO

RECEPÇÃO



Canal de transmissão

Comunicação Serial

Direção da Transferência dos Dados

a) SIMPLEX



b) HALF-DUPLEX



c) FULL-DUPLEX



Comunicação Serial

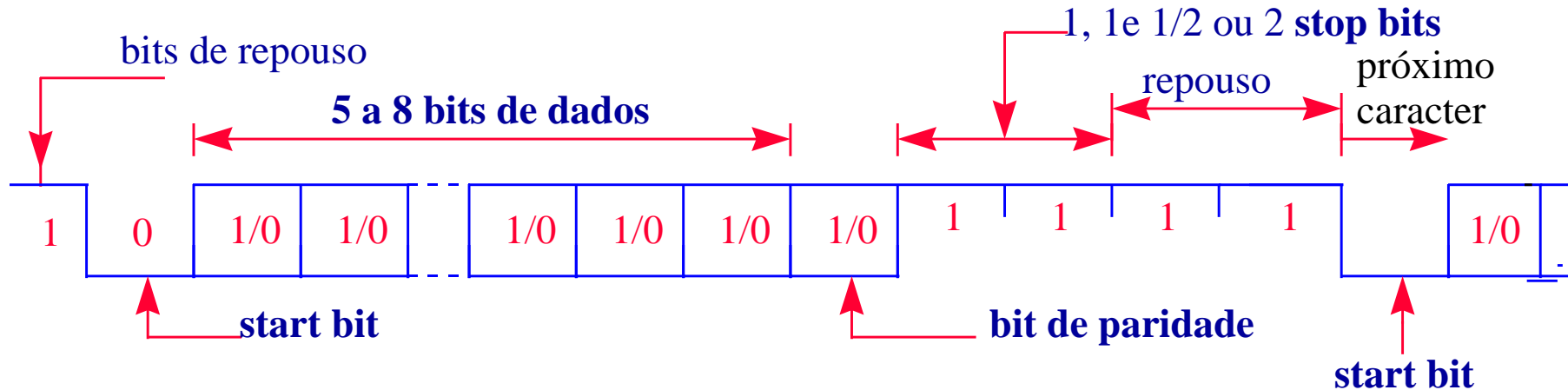
Tipos de Comunicação Serial

Síncrona: um sinal de clock em separado é associado com o dado.

Assíncrona: não existe sincronismo entre transmissor e receptor - a re-sincronização é feita caractere por caractere.

Comunicação Serial Assíncrona

a. Formato do Caractere



b. Sincronismo

O sincronismo é feito a cada caractere, através de um clock localizado no receptor, cujo período é **K** vezes menor do que o período de duração do bit.

$$TB = K.Tc$$

TB - tempo de duração do bit

Tc - período do clock de sincronismo do caractere

Comunicação Serial Assíncrona

e. Velocidade de Comunicação

- Considerando um caractere formado por 11 bits, como segue:
 - 1 **start** bit
 - 8 bits de dados
 - 1 bit de paridade
 - 1 **stop** bit
- Considerando por exemplo, uma velocidade de comunicação de 10 caracteres por segundo, tem-se:

$$11 \text{ bits/caractere} \times 10 \text{ caracteres/s} = 110 \text{ bits/s}$$

- Define-se:

$$\text{BAUD} = \text{bits/s}$$

$$\text{BAUD RATE} = \text{Taxa de Comunicação}$$

Comunicação Serial Assíncrona

f. Taxas de Comunicação mais Comuns

Taxa	TB
110 Baud	9.1 ms
150 Baud	6.66 ms
300 Baud	3.33 ms
600 Baud	1.66 ms
1200 Baud	833 us
2400 Baud	416 us
4800 Baud	208 us
9600 Baud	104 us
19200 Baud	52 us

Comunicação Serial RS 232 - C

Padronizada pela **EIA** (**E**lectronic **I**ndustries **A**ssociation - USA)

RS : Recommended Standard

232 : número da norma

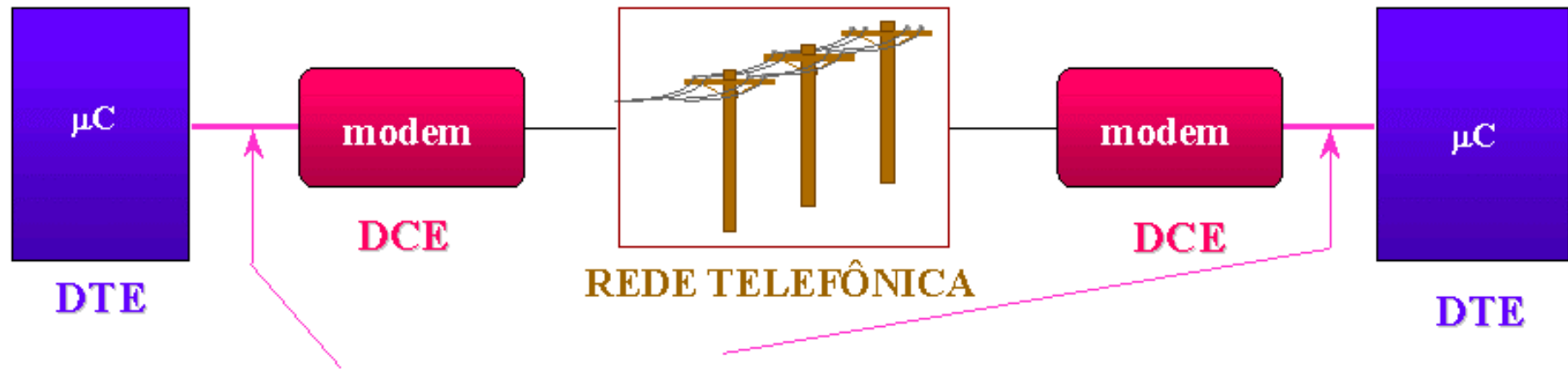
C : número de revisões da norma

Tipo de comunicação : Serial Assíncrona

Características da Interface

- ★ Taxa de Comunicação de 75 Baud a 19200 Baud
- ★ Comprimento do cabo de ligação entre equipamentos: menor do que 15 metros, sem amplificação.
- ★ Não existe isolação elétrica entre os equipamentos

Comunicação Serial RS 232 - C



Um **Protocolo de Comunicação de Dados** estabelece regras para a comunicação entre um **DTE** e um **DCE** ou entre um **DTE** e um **DTE**.

DTE : Data Terminal Equipment

Equipamento que compreende a fonte de dados, a recepção de dados ou ambos

DCE : Data Communication Equipment

Equipamento que providencia as funções necessárias para manter uma conexão de dados, e realizar a conversão do sinal necessária à comunicação entre um DTE e o circuito de comunicação.

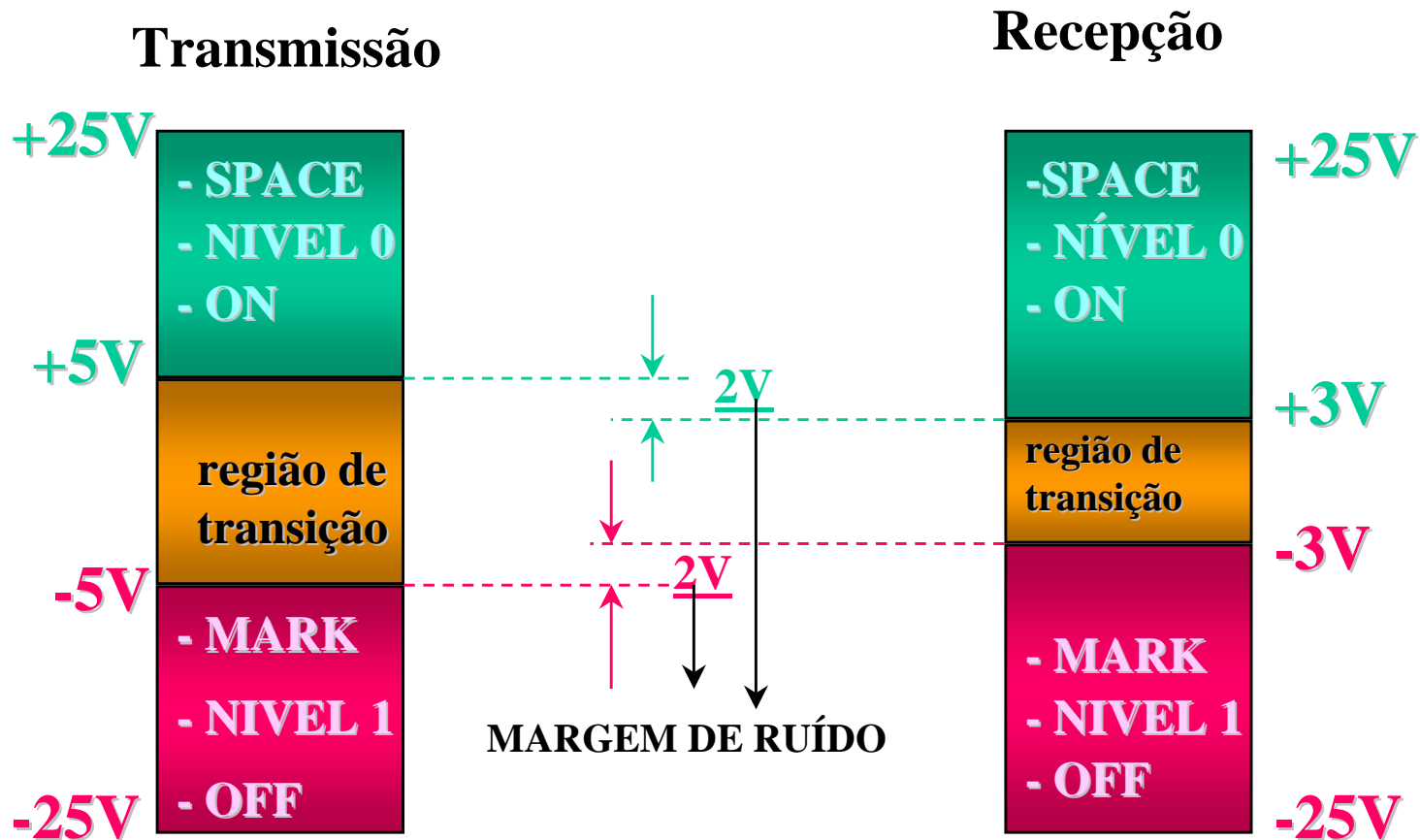
Comunicação Serial RS 232 - C

Características Elétricas do Sinal

★ Lógica Negativa

STATUS		
nível lógico	1	0
nível de tensão	- 25V a -3V	+ 3 V a + 25V
função	OFF	ON
condição do sinal	MARK	SPACE

Comunicação Serial RS 232 - C



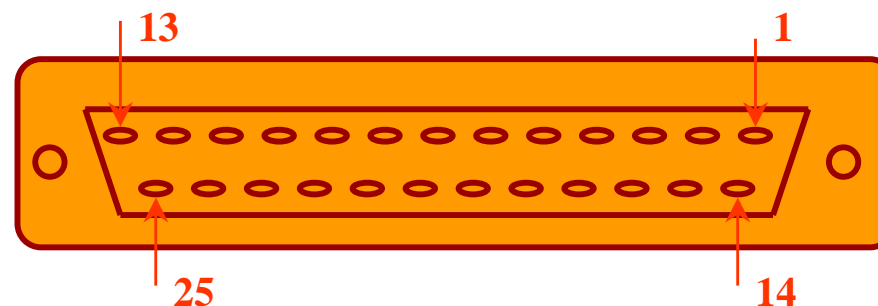
Comunicação Serial RS 232 - C

Características Mecânicas da Interface

- Conector de 25 pinos - **DB 25**

- **DB 25 P (macho)** - **DTE**
- **DB 25 S (fêmea)** - **DCE**

Conjunto de Sinais para o DTE



PINO	NOME	FUNÇÃO	PINO	NOME	FUNÇÃO
1	GND	Terra da carcaça	13	SCTS	CTS secundário
2	TXD	Transmissão de dados	14	STxD	Transmissão secundário
3	RXD	Recepção de dados	15	TSET	Transmit Signal Element Timing
4	RTS	Request To Send	16	SRxD	Recepção secundário
5	CTS	Clear To Send	17	RSET	Receive Signal Element Timing
6	DSR	Data Set Ready	18	-	Não usado
7	GND	Terra do sinal	19	SRTS	RTS secundário
8	DCD	Data Carrier Detect	20	DTR	Data Terminal Ready
9/10	-	reservado para testes no conjunto de dados	21	SQD	Signal Quality Detector
11	-	Não usado	22	RI	Ring Indicator
12	SDCD	DCD secundário	23	DSRS	Data Signal Rate Select
			24	TSET	Transmit signal Element Timing
			25	-	Não usado

Comunicação Serial RS 232 - C

Características Funcionais do Circuito

Transmissão de Dados (TxD)



Esta linha permanece em nível lógico “1” (tensão negativa), enquanto **não** tiver dados.

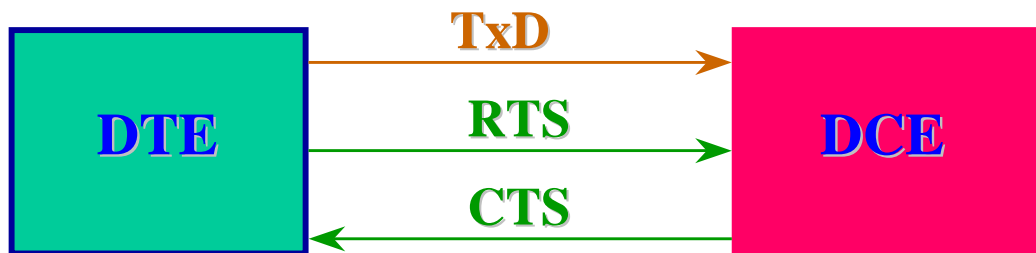
Recepção de Dados (TxD)



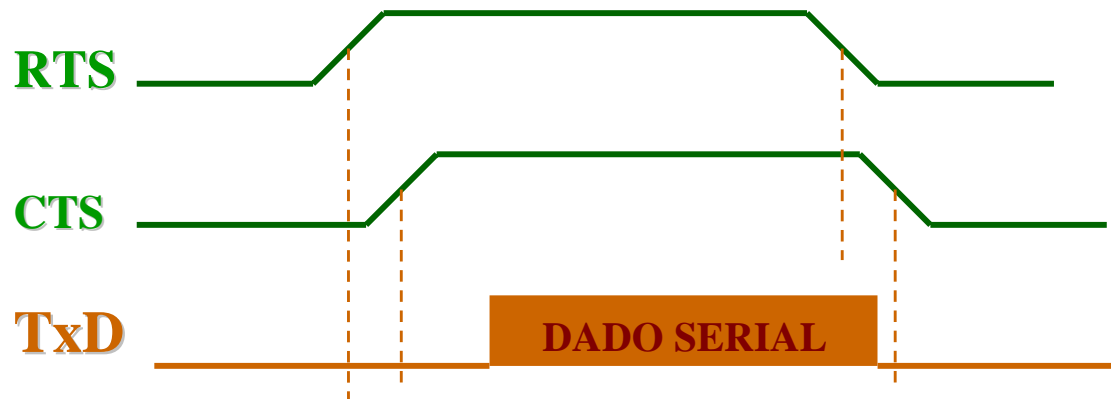
Esta linha permanece em nível lógico “1” (tensão negativa), enquanto **não** tiver dados.

Comunicação Serial RS 232 - C

Comunicação Serial com “Handshaking”

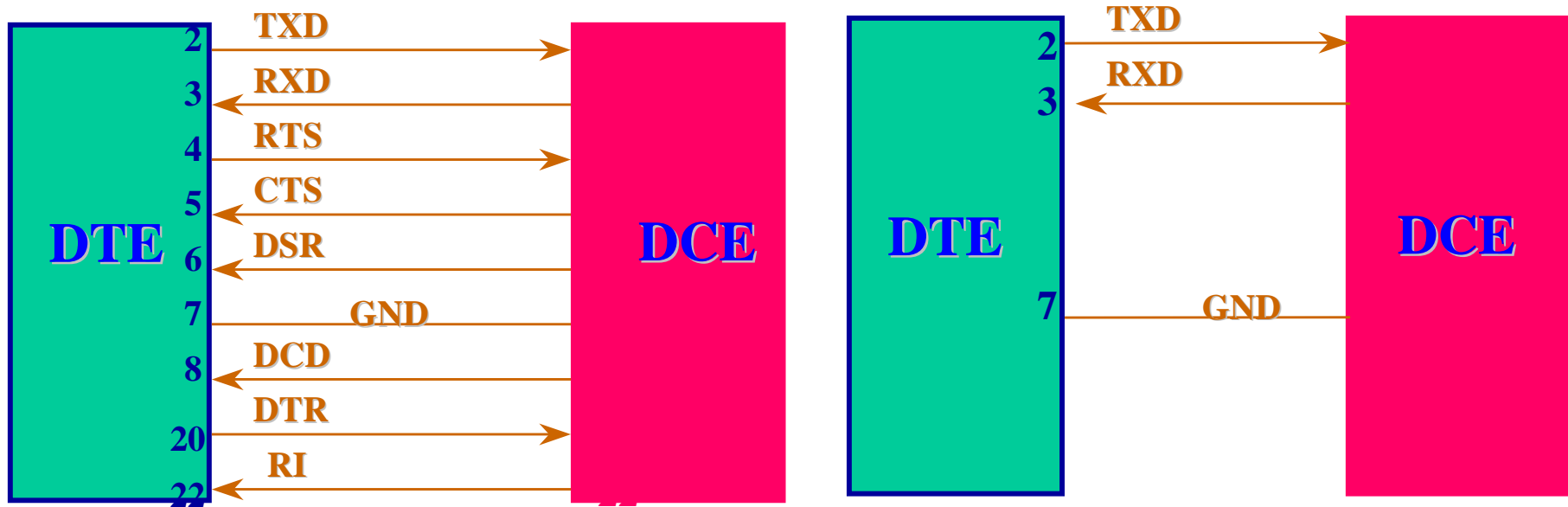


RTS – Request to Send
CTS – Clear to Send



Comunicação Serial RS 232 - C

Implementação em Microcomputadores

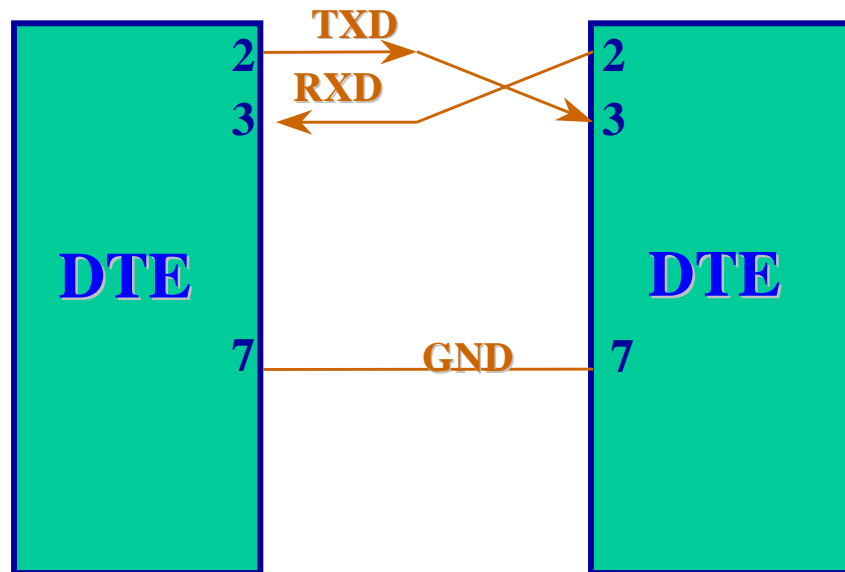


Full Duplex Padrão

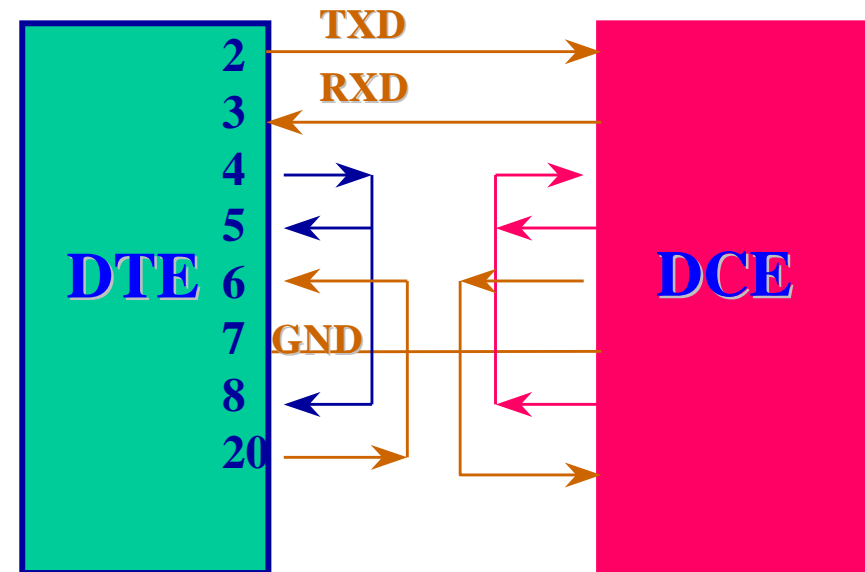
Três Fios

Comunicação Serial RS 232 - C

Modem Nulo



Três Fios com Loop de Luxo



pino 4: RTS
pino 5: CTS
pino 6: DSR

pino 8: DCD
pino 20: DTR

Interface Serial no Microcontrolador 8051

A porta serial é full-duplex

- Para Transmitir um Dado Serialmente Escrever no SBUF

- Para Receber um Dado Serialmente Ler do SBUF

F8							
F0	B						
E8							
E0	ACC						
D8							
D0	PSW						
C8	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	
C0							
B8	IP						
B0	P3						
A8	1E						
A0	P2						
98	SCON						
90	P1						
88	TCON	TMOD	TL0	TL1	TH0	TH1	
80	P0	SP	DPL	DPH			PCON

- Os registradores **SBUF** de recepção e de transmissão são fisicamente separados.

Interface Serial no Microcontrolador 8051

SCON → Usado para programar a Interface Serial

Endereçável a Bit

F8								
F0	B							
E8								
E0	ACC							
D8								
D0	PSW							
C8	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2		
C0								
B8	IP							
B0	P3							
A8	1E							
A0	P2							
98	SCON	SBUF						
90	P1							
88	TCON	TMOD	TL0	TL1	TH0	TH1		
80	P0	SP	DPL	DPH				PCON

PCON → Possui um bit (SMOD) que permite dobrar Baud Rate

Não endereçável a Bit

Interface Serial no Microcontrolador 8051

A porta serial pode operar em 4 Modos

Modo 0 – Modo Síncrono: O dado serial (de 8 Bits com LSB primeiro) é transmitido e recebido através de RxD. O TxD envia o clock. A Taxa de Comunicação (“baud rate”) é fixa em 1/12 da frequência do oscilador .

Modo 1 – Modo Assíncrono: 10 Bits são transmitidos (via TxD) ou recebidos (via RxD) : 1 StartBit, 8 Bits de dados com LSB primeiro e um StopBit. O “baud rate” é variável .

Modo 2 – Modo Assíncrono: 11 Bits são transmitidos (via TxD) ou recebidos (via RxD) : 1 StartBit, 8 Bits de dados com LSB primeiro, um nono Bit programável, e um StopBit . O “baud rate” é programável para 1/32 ou 1/64 da frequência do oscilador.

Modo 3 – Modo Assíncrono: 11 Bits são transmitidos (via TxD) ou recebidos (via RxD) : 1 StartBit, 8 Bits de dados com LSB primeiro, um nono Bit programável, e um StopBit. O “baud rate” é variável .

Programação dos Modos de Operação da Interface Serial

Registrador de Controle da Porta Serial - **SCON** – Endereçável a Bit

SCON

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0	SM1	Modo de Funcionamento	Taxa de Transmissão
0	0	0	Fclock / 12
0	1	1	Variável (10 bits)
1	0	2	Fclock / 32 ou Fclock / 64
1	1	3	Variável (11 bits)

SM2 - Habilita a comunicação em multiprocessadores nos modos 2 e 3 .
No modo 0 , SM2 deve ser igual a zero .

Programação dos Modos de Operação da Interface Serial

SCON

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
------------	------------	------------	------------	------------	------------	-----------	-----------

REN - Habilita a recepção serial → 1 Habilita a recepção Serial
→ 0 Desabilita a recepção Serial

TB8 - É o nono bit de dados que será transmitido nos modos 2 e 3. Setado ou zerado por Software.

RB8 - É o nono bit de dado que foi recebido nos modos 2 e 3. No Modo 1, se $SM2=0$ → RB8 é o StopBit recebido. No Modo 0 RB8 não é usado.

TI - Flag de Interrupção da Transmissão. Setado por Hardware no fim do oitavo bit no Modo 0, ou no começo do StopBit nos outros Modos.

RI - Flag de Interrupção da Recepção. Setado por Hardware no fim do oitavo bit no Modo 0, ou na metade do tempo do StopBit, nos outros Modos.

(Baud Rates) Taxas de comunicação

Modo 0 : Baud Rate fixo. $\rightarrow = \text{Freqüência do oscilador} / 12$

Modo 2 : Baud Rate fixo. $\rightarrow = \text{Freqüência do oscilador} / 32 \text{ ou } 64$

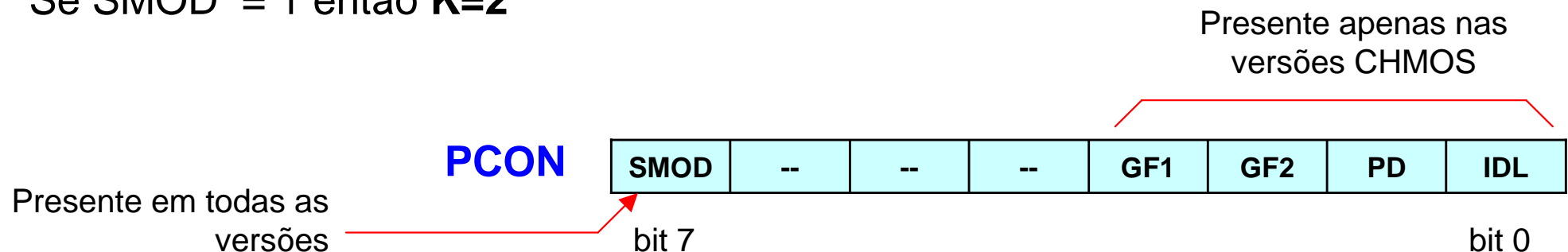
Modo 1 ou Modo 3 : Baud Rate variável. \rightarrow Pode ser gerado tanto pelo Timer 1 ou Timer 2 (8052)

O T1 é usado no Modo 2 (Auto Reload) como Timer (C/T = 0)

$$\text{BaudRate} = \frac{K \times (\text{Freqüência do Oscilador})}{32 \times 12 \times [256 - (TH1)]}$$

Se SMOD = 0 então **K=1** (default)

Se SMOD = 1 então **K=2**



(Baud Rates) Taxas de comunicação

- Logo, deve-se calcular o valor de TH1 (Byte mais significativo do Contador 1, que no Modo 2 é carregado em TL1 no fim de cada contagem)
- TH1 é um valor inteiro de 8 Bits (de 00 a 255)

$$TH1 = 256 - \frac{K \times (\text{Frequência do Oscilador})}{384 \times (\text{Baud Rate})}$$

- Arredonda-se TH1 para o inteiro mais próximo.
- Como o arredondamento pode não produzir o Baud Rate desejado, deve-se escolher uma outra frequência para o cristal adotando-se o valor arredondado de TH1.

Exemplo:

- Gerar a Taxa de Comunicação de 19,2 KBPS (19.200 BPS) sendo a freqüência do cristal de 12 MHz.

$$TH1 = 256 - \frac{K \times (\text{Freqüência do Oscilador})}{384 \times (\text{Baud Rate})}$$

Fazendo SMOD = 1 então K = 2

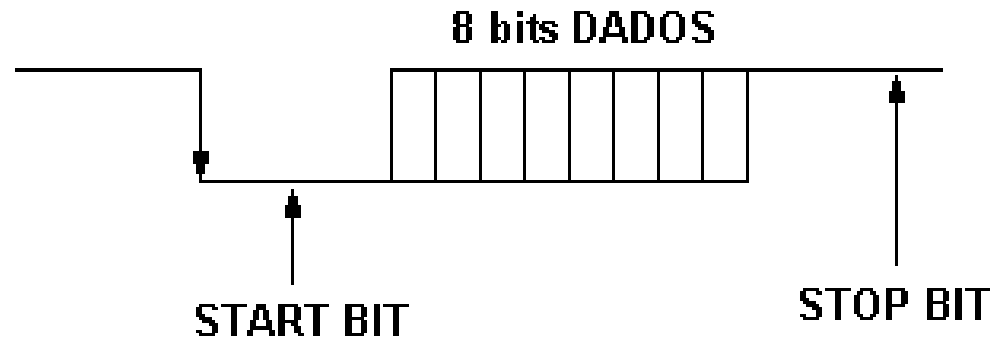
$$TH1 = 256 - \frac{2 \times 12 \times 10^6}{384 \times 19200} = 252,74 \cong 253 \Rightarrow 0FDh$$

- Como TH1 deve ser inteiro, deve-se ajustar a freqüência do cristal :

$$\text{Freqüência do Oscilador} = \frac{\text{BaudRate} \times 384 \times (256 - TH1)}{K}$$

$$\text{Freqüência do Oscilador} = \frac{19200 \times 384 \times (256 - 253)}{2} = 11059200 \text{ Hz} = 11.059 \text{ MHz}$$

Operação no Modo1: Comunicação serial Assíncrona. (10 Bits)



Transmissão :

1. A transmissão começa escrevendo-se o dado em **SBUF**

```
MOV SBUF,#dado
```

2. Quando o MSB for transmitido, o bit **TI** de **SCON** é setado.

Operação no Modo1: Comunicação serial Assíncrona. (10 Bits)

Recepção :

- 1. Iniciada quando uma transição de 1 para 0 é detectada em RxD . A linha é amostrada 16 vezes. O valor é aceito após 2 ou 3 amostras para evitar ruído (falso StartBit)**
- 2. O dado recebido é deslocado por um Shift Register e ao final carregado em SBUF, o StopBit em RB8 e RI é setado .**
- 3. Isto só ocorre se a seguinte condição existir no final da última recepção**
 - a) $Ri = 0$**
 - b) SM2 ou StopBit recebido = 1**

Se estas condições não ocorrerem, o dado foi perdido

Exemplo: Transmitir o código ASCII da letra A pelo canal serial a um taxa de 1200 BPS.

Considerar o cristal da CPU de 11,0592 MHz.

Timer 1				Timer 0			
GATE	C / T	M1	M0	GATE	C / T	M1	M0

```

ORG 0
MOV TMOD,#20h           ;TMOD = 00100000 →Timer 1 no
                        ;Modo 2, controle por software
                         $TH1 = 256 - \frac{K \times (Frequência\ do\ Oscilador)}{384 \times (Baud\ Rate)}$ 
MOV TH1,#232           ;valor 232 em TH1 e TL1 para
                        ;gerar a Taxa de Comunicação de
MOV TL1,#232           ;1200 BPS com fc=11,0592 MHz e
                        ; K=1(default)
SETB TR1               ;Dispara Temporizador
MOV SCON,#40h          ;SCON = 01000000 → modo 1 do
                        ;Canal Serial
                        

|     |     |     |     |     |     |    |    |
|-----|-----|-----|-----|-----|-----|----|----|
| SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
|-----|-----|-----|-----|-----|-----|----|----|


MOV SBUF, #'A'         ; Transmite o caractere ASCII A
JNB TI,$               ; Espera terminar a transmissão
CLR TI                 ; Prepara para nova transmissão
SJMP $                 ; Pára

```

Exemplo: Receber um código ASCII pelo canal serial a um taxa de 1200 BPS.

Considerar o cristal da CPU de 11,0592 MHz.

	SCON	<table border="1"><tr><td>SM0</td><td>SM1</td><td>SM2</td><td>REN</td><td>TB8</td><td>RB8</td><td>TI</td><td>RI</td></tr></table>	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SM0	SM1	SM2	REN	TB8	RB8	TI	RI			
<i>ORG 0</i>										
<i>MOV TMOD,#20h</i>		;TMOD = 00100000 → Timer 1 no ;Modo 2, controle por software								
<i>MOV TH1,#232</i>		;valor 232 em TH1 e TL1 para ;gerar a Taxa de Comunicação de								
<i>MOV TL1,#232</i>		;1200 BPS com $f_c=11,0592$ MHz e ; K=1(default)								
<i>SETB TR1</i>		;Dispara Temporizador								
<i>MOV SCON,#40h</i>		;SCON = 01000000 → modo 1 do ;Canal Serial								
<i>SETB REN</i>		; Habilita a Recepção								
<i>JNB RI,\$</i>		;Espera terminar a recepção								
<i>MOV A, SBUF</i>		; Lê o dado recebido serialmente								
<i>CLR RI</i>		; Prepara para nova recepção								
<i>SJMP \$</i>		; Pára								

Exemplo: Receber e Transmitir qualquer caractere ASCII pelo canal serial a um taxa de 1200 BPS, utilizando a Interrupção.

Considerar o cristal da CPU de 11,0592 MHz.

```
FLAG      ORG 0
          EQU 30H          ; Definição de Flag de Atendimento de Interrupção
          SJMP PROG

          ORG 0023H       ; Sub-rotina de atendimento da Interrupção Serial
          CLR EA          ; Desabilita as interrupções
          JNB TI,FIM      ; Verifica se foi Recepção
          CLR TI          ; Sendo Transmissão, limpa o Flag TI de Transmissão
          SJMP FIM1       ; Retorna
FIM:      MOV A,SBUF      ; Lê um caractere Serial
          CLR RI          ; Sendo Recepção, limpa o Flag RI de Recepção
FIM1:     SETB EA         ; Re-abilita as interrupções e
          SETB FLAG       ; Ativa Flag de Atendimento de Interrupção
          RETI           ; retorna

PROG:     SETB EA         ; Habilitação de Interrupções
          SETB IE.4       ; Habilitação da Interrupção Serial
          MOV TMOD,#20H   ; TMOD = 00100000 - Timer 1 no Modo 2, controle por software
          MOV TH1,#232    ; valor 232 em TH1 e TL1 para gerar a Taxa de Comunicação de
          MOV TL1,#232    ; 1200 BPS com fc=11,0592 MHz e K=1(default)
          SETB TR1        ; Dispara Temporizador
          MOV SCON,#40h   ; SCON = 01000000 ? modo 1 do Canal Serial
          SETB REN        ; Habilita a Recepção
LOOP:    CLR FLAG        ; Zera o Flag de Atendimento de Interrupção
          JNB FLAG,$      ; Verifica se já foi atendida a Interrupção Serial
          MOV SBUF,A      ; Transmite o caractere recebido
          CLR FLAG        ; Zera o Flag de Atendimento de Interrupção
          JNB FLAG,$      ; Verifica se já foi atendida a Interrupção
          SJMP LOOP       ; Continua a comunicação Serial (Recebe/Transmite)
```

Exercício:

Construir um programa que seja capaz de enviar pela porta serial uma frase armazenada na memória de programa do microcontrolador, toda vez que o número 50h (código ASCII da letra “P”) for recebido pela porta serial.

Utilizar Baud Rate de 9600.

Construir em duas versões:

- 1 - sem o uso da interrupção serial;**
- 2 – utilizando interrupção serial.**