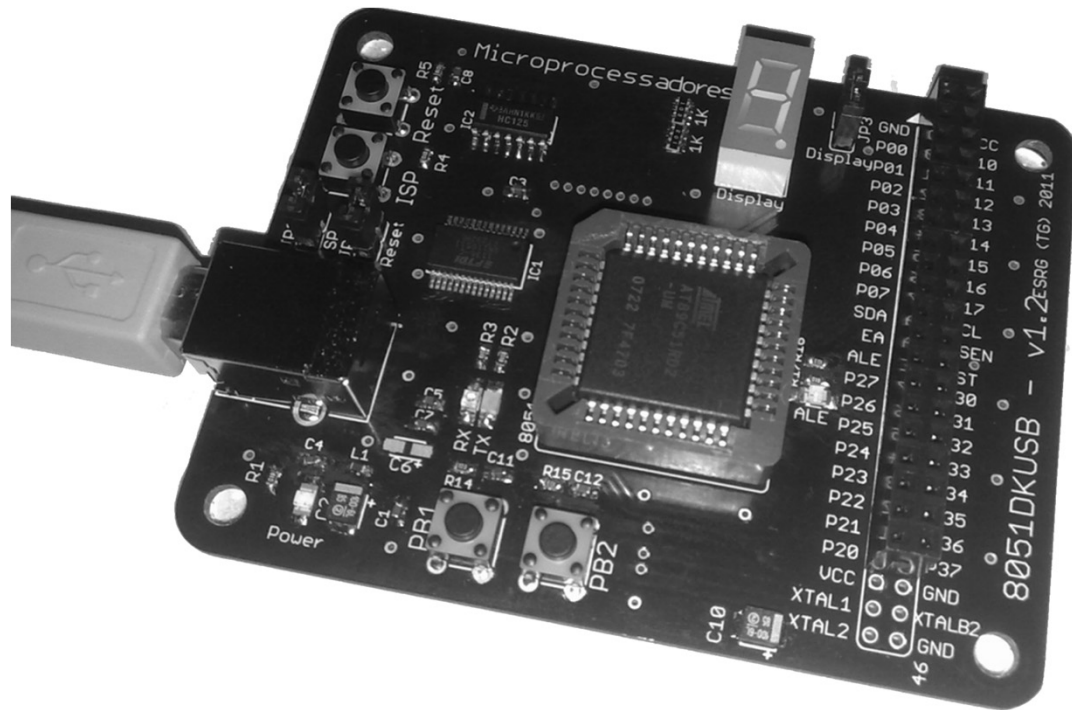
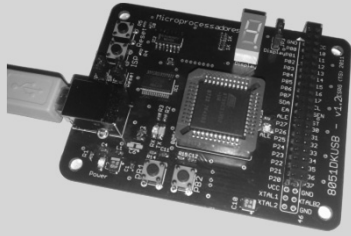


# Mestrado Integrado em Eng. Electrónica Industrial e Computadores



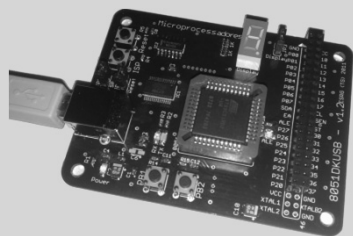
Comunicações série

Microcontroladores  
2º Ano – A10



# ***Comunicação: paralela x série***

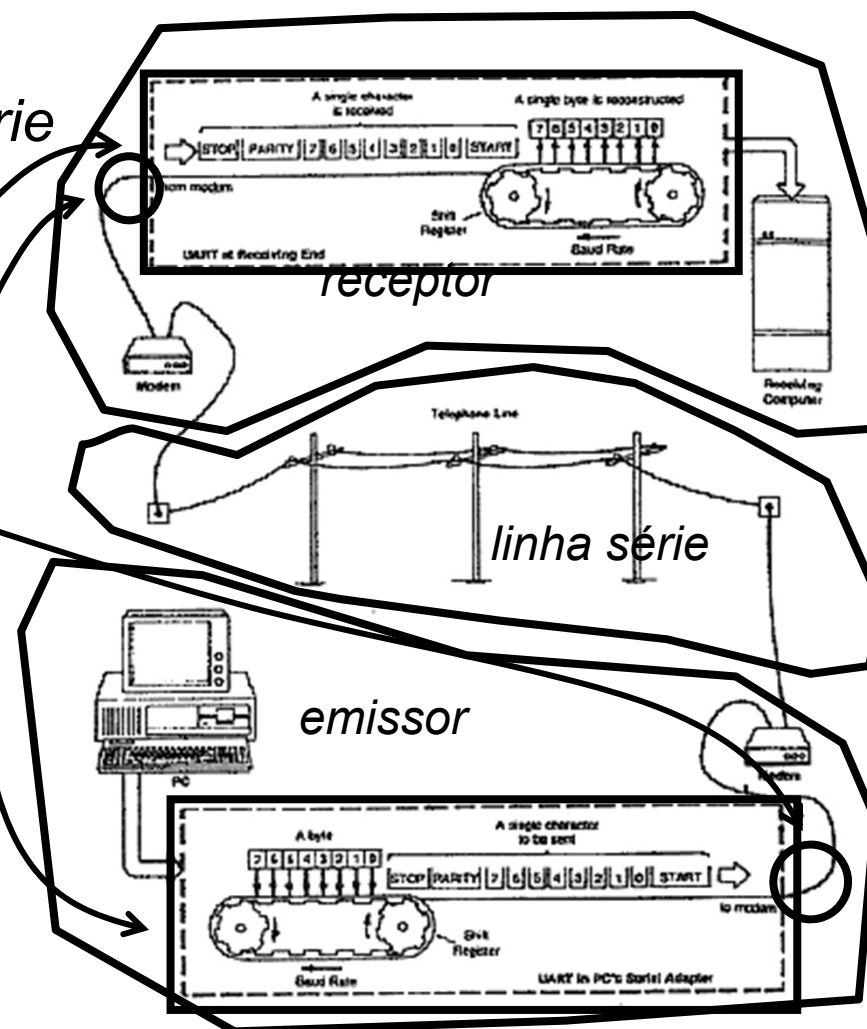
- *Comunicação paralela:*
  - *Rápida, mas limitada na distância*
  - *Economicamente cara*
- *Comunicação série:*
  - *Grandes distâncias*
  - *Solução mais económica*

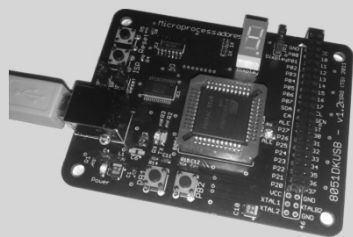


# Comunicação série

- Componentes na comunicação série

- Conversor paralelo-série
- Conversor série-paralelo
- Conversor analógico-digital
- Conversor digital-analógico

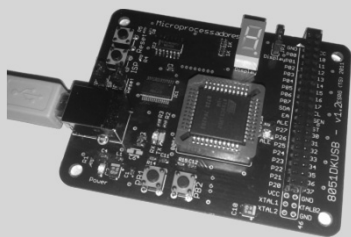




# Comunicação série

## Assíncrona x Síncrona

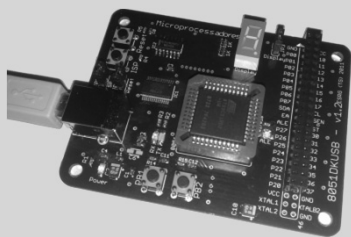
- *A transferência dos dados pode ser efectuada usando dois métodos distintos:*
  - *Comunicação síncrona: orientada a transferência de blocos de dados em cada unidade de transferência*
    - *Nota: Na comunicação síncrona geralmente é usado uma linha própria com o sinal de relógio ou este é extraído do próprio sinal através de técnicas de processamento de sinal*
  - *Comunicação assíncrona: orientada a transferência de um único byte de dado em cada unidade de transferência*
    - *Nota: Na comunicação assíncrona o sincronismo é extraído pela análise do pacote de dados*



# Comunicação série

## Assíncrona x Síncrona

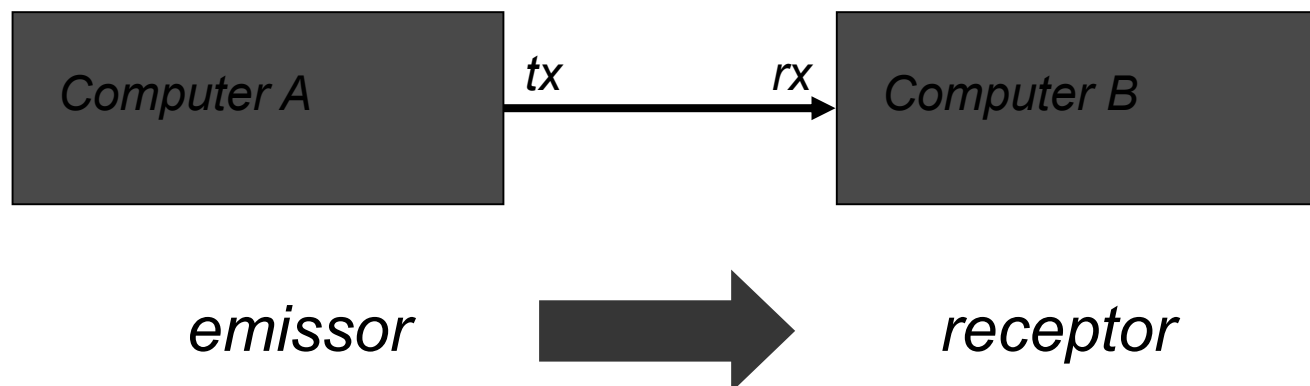
- *O controlo da transferência de dados é feita por hardware especializado - controladores de comunicação série:*
  - *USART (Universal Synchronous-Asynchronous Receiver-Transmitter) – comunicação síncrona e assíncrona*
  - *UART (Universal Asynchronous Receiver-Transmitter) – comunicação assíncrona*

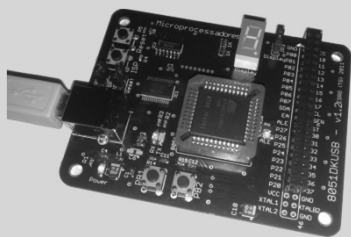


# Comunicação série

## Tipos de transmissão

- *Existem três tipos de comunicação:*
  - *Simplex: A comunicação apenas é efectuada num dos sentidos*

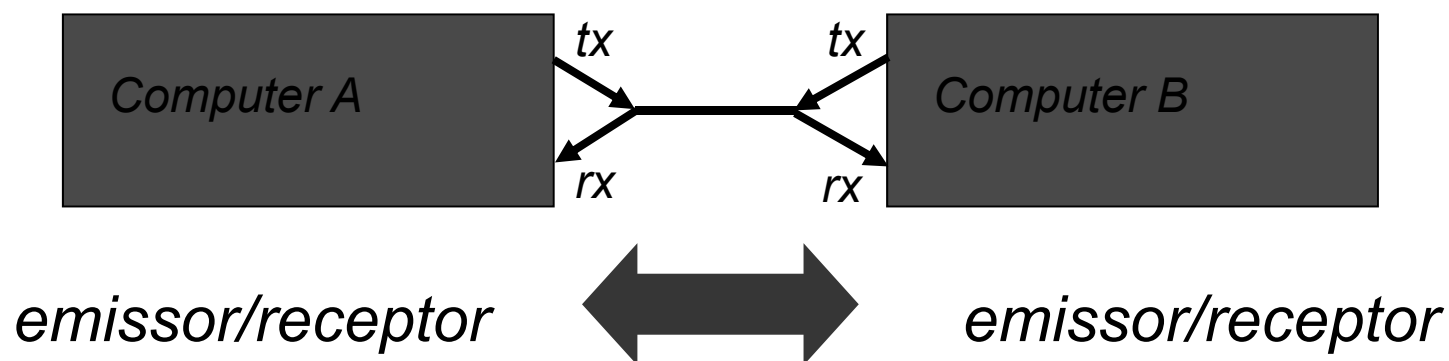


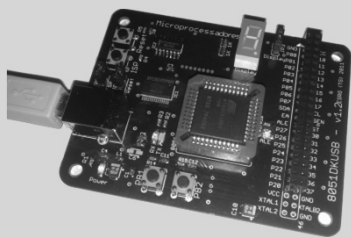


# Comunicação série

## Tipos de transmissão

- *Half-duplex: A comunicação é efectuada nos dois sentidos, mas apenas num sentido de cada vez*

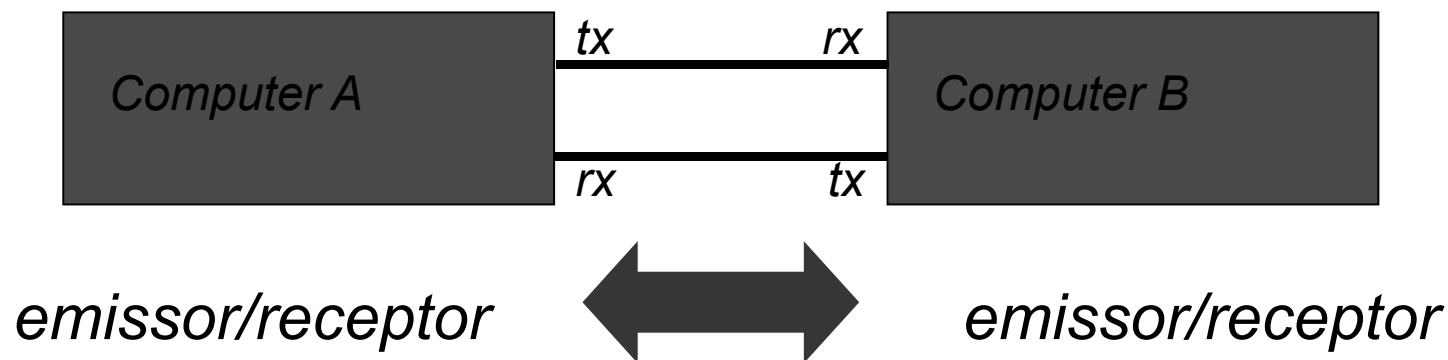




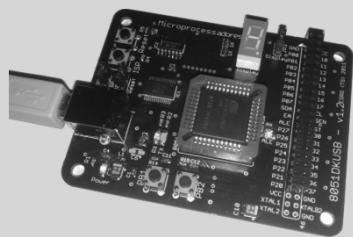
# Comunicação série

## Tipos de transmissão

- *Full-duplex: A comunicação pode ser efectuada nos dois sentidos simultaneamente*



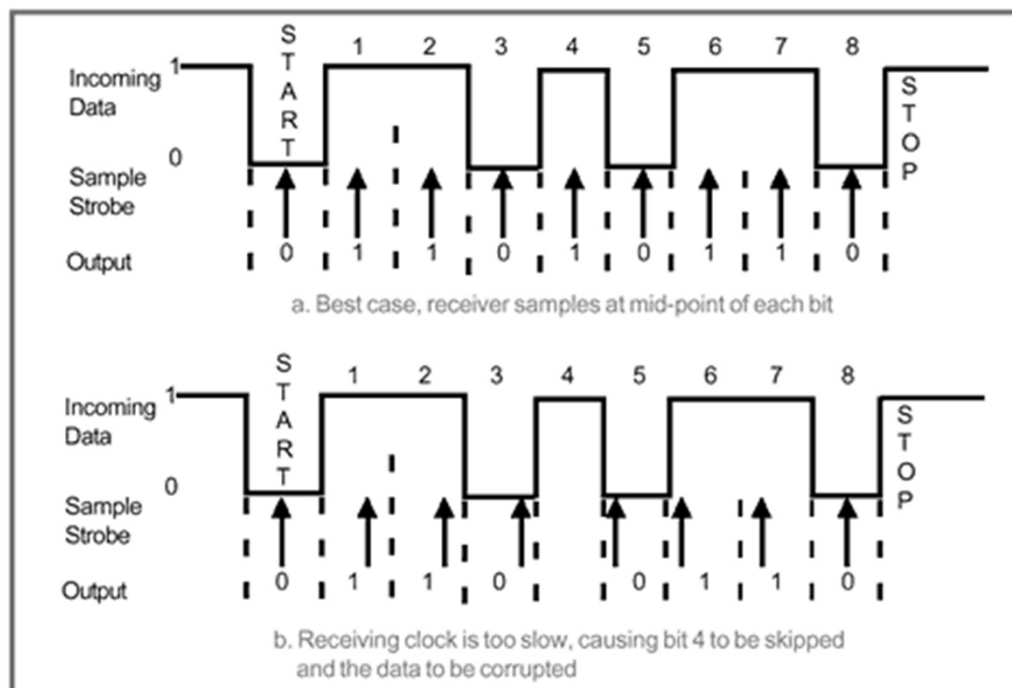




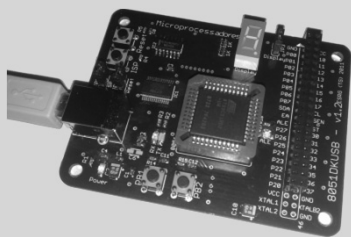
# Comunicação série

## Estrutura da “frame”

- Os bytes durante a transmissão são empacotados em frames
- A estrutura da frame consiste num start bit, nos bits de dados, num bit de paridade (opcional) e num stop bit.



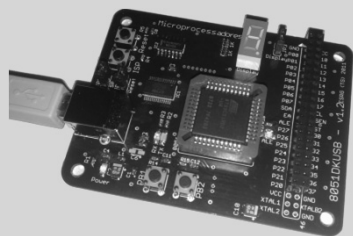
- O start bit é amostrado a uma frequência superior (16 vezes) para determinar o meio do bit
- O ponto intermédio do bit é usado posteriormente para determinar o valor dos outros bits



# Comunicação série

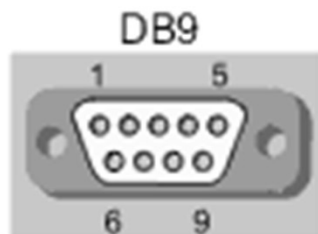
## Interface física

- *A interface física geralmente usada é denominada de RS-232*
- *Esta interface define os seguintes níveis eléctricos:*
  - *'0':  $> +3V$ , tipicamente  $+12V$*
  - *'1':  $< -3V$ , tipicamente  $-12V$*
- *Está tipicamente limitada a velocidade inferiores a 100.000 baud (depende do controlador de comunicações)*
- *Baud é a unidade da velocidade de transmissão de dados, equivalente ao números de bits por segundo. Também podemos indicar a unidade de transferência como sendo 'bps' (bits per second)*

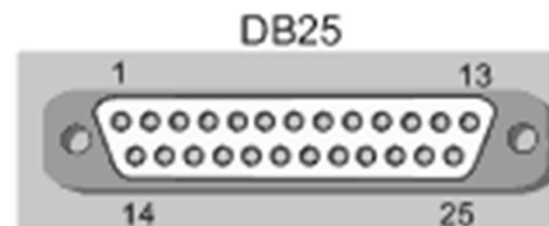


# Comunicação série

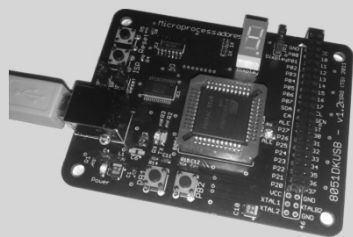
## Fichas de conexão



Pin	Signal
1	Received Line Signal Detector (DCD)
2	Received Data (RxD)
3	Transmitted Data (TxD)
4	Data Terminal Ready (DTR)
5	Signal Ground
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)
Shell	Frame Ground



Pin	Signal
1	Frame Ground
2	Transmitted Data (TxD)
3	Received Data (RxD)
4	Request to Send (RTS)
5	Clear to Send (CTS)
6	Data Set Ready (DSR)
7	Signal Ground
8	Received Line Signal Detector (DCD)
20	Data Terminal Ready (DTR)
22	Ring Indicator (RI)
Shell	Frame Ground

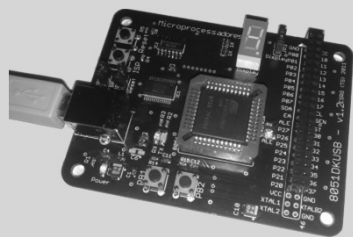


# Comunicação série - 8051

- O porto série no 8051 funciona no modo full duplex
  - *Recepção e transmissão simultânea*
- *A recepção é feita pela leitura do registo SBUF, enquanto o envio é efetuado pela escrita no mesmo registo*

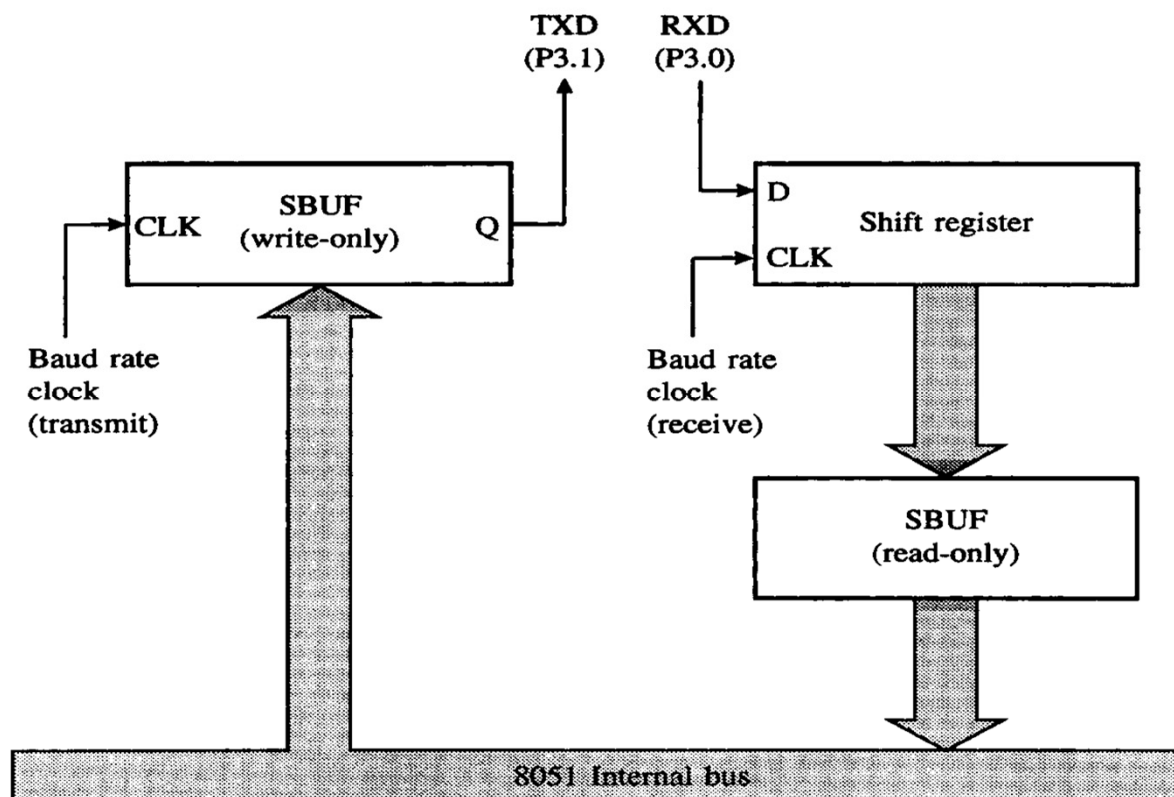
Ex.      – recepção:            `mov A, SBUF`  
          – emissão:            `mov SBUF, A`

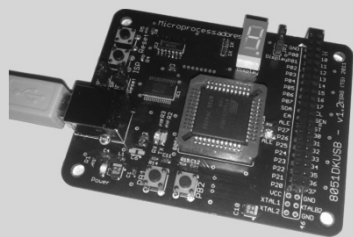
- *A recepção é “buffered”, ou seja existe um registo temporário no “shift register” de recepção que guarda os bits do byte até este ter sido totalmente recebido*
  - *Após a recepção de todos os bits do byte, se houver um byte no registo SBUF que ainda não tenha sido lido, este será destruído*



# Comunicação série - 8051

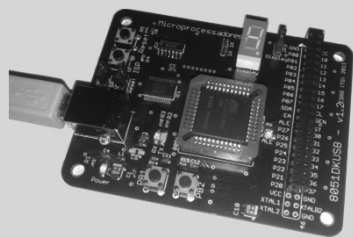
- Diagrama de bloco genérico da porta série





# Comunicação série - 8051

- **Interface série no microcontrolador 8051**
- ***Registos usados na programação da comunicação série:***
  - *SCON: Permite programar os modos de funcionamento e o estado da comunicação série.*
  - *SBUF: Permite ler e escrever dados no porto série.*
  - *[IE: Permite controlar a geração da interrupção associada a comunicação série]*
  - *Registos de programação dos timers*



# Comunicação série - 8051

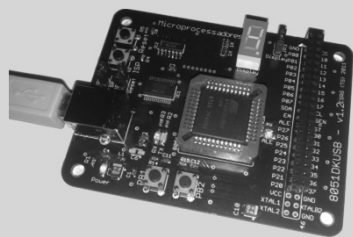
## • Interface série no microcontrolador 8051

- Registo: **SCON**

MSB				LSB			
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0, SM1:		Mode:	Description:	Baud Rate:
0	0	0	shift register	$f_{osc}/12$
0	1	1	8-bit UART	variável
1	0	2	9-bit UART	$f_{osc}/32, f_{osc}/64$
1	1	3	9-bit UART	variável

**SM2:** Activa a capacidade de comunicação entre múltiplos processadores existente nos modos 2 e 3. No modo 2 e 3, se  $SM2=1$  então  $RI$  não será activado se o nono bit em  $RB8$  for 0. No modo 1, se  $SM2=1$  então  $RI$  não será activado se não for encontrado um stop bit válido. No modo 0,  $SM2$ , deve ser 0.



# Comunicação série - 8051

- Interface série no microcontrolador 8051

- Registo: **SCON**

MSB				LSB			
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

*REN: Controla activação da recepção série. Manipulado por software*

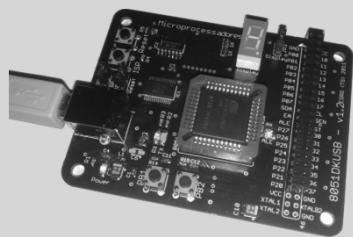
*TB8: Nono bit que será transmitido nos modos 2 e 3*

*RB8: Nono bit recebido nos modos 2 e 3. No modo 1, se SM2= 0, RB8 é o stop bit. No modo 0, este bit não é usado*

*TI: Flag de interrupção da transmissão. Activada por hardware ao fim do 8º bit no modo 0, ou no começo do stop bit nos outros modos. Tem que ser apagado por software*

*RI: Flag de interrupção da recepção. Activada por hardware ao fim do 8º bit no modo 0, ou no meio do stop bit nos outros modos (depende de SM2). Tem que ser apagado por software*





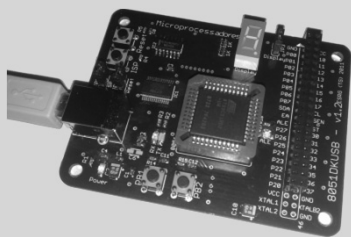
# Comunicação série - 8051

- **Interface série no microcontrolador 8051**

*O porto série pode operar em quatro modos:*

- ***Modo 0:***

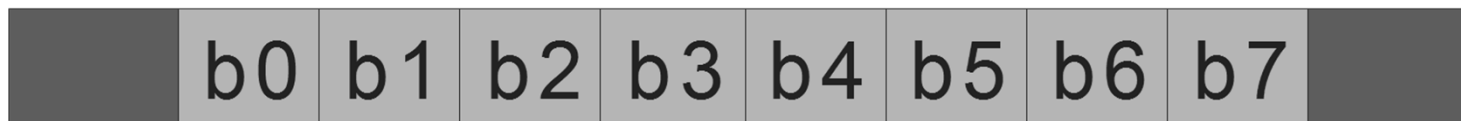
- *Os dados saem e entram através do pino P3.0 (RxD).*
- *O pino P3.1 (TxD) é usado para enviar o sinal de sincronismo para extracção dos bits*
- *São enviados/recebidos 8 bits de cada vez (LSB é enviado primeiro)*
- *O baud rate é de 1/12 da frequência do oscilador*



# Comunicação série - 8051

- Interface série no microcontrolador 8051

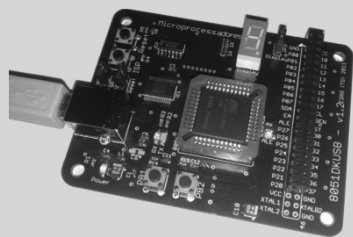
RxD



TxD

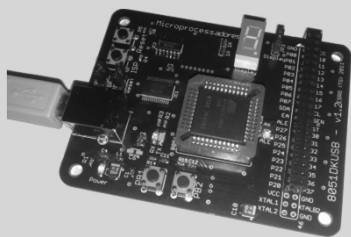


- *Os bits do porto série são geralmente amostrados a 16x o valor do baud rate*
- *No modo 0, como o baud rate é demasiado elevado, é necessário enviar o sinal para extracção dos bits separadamente*



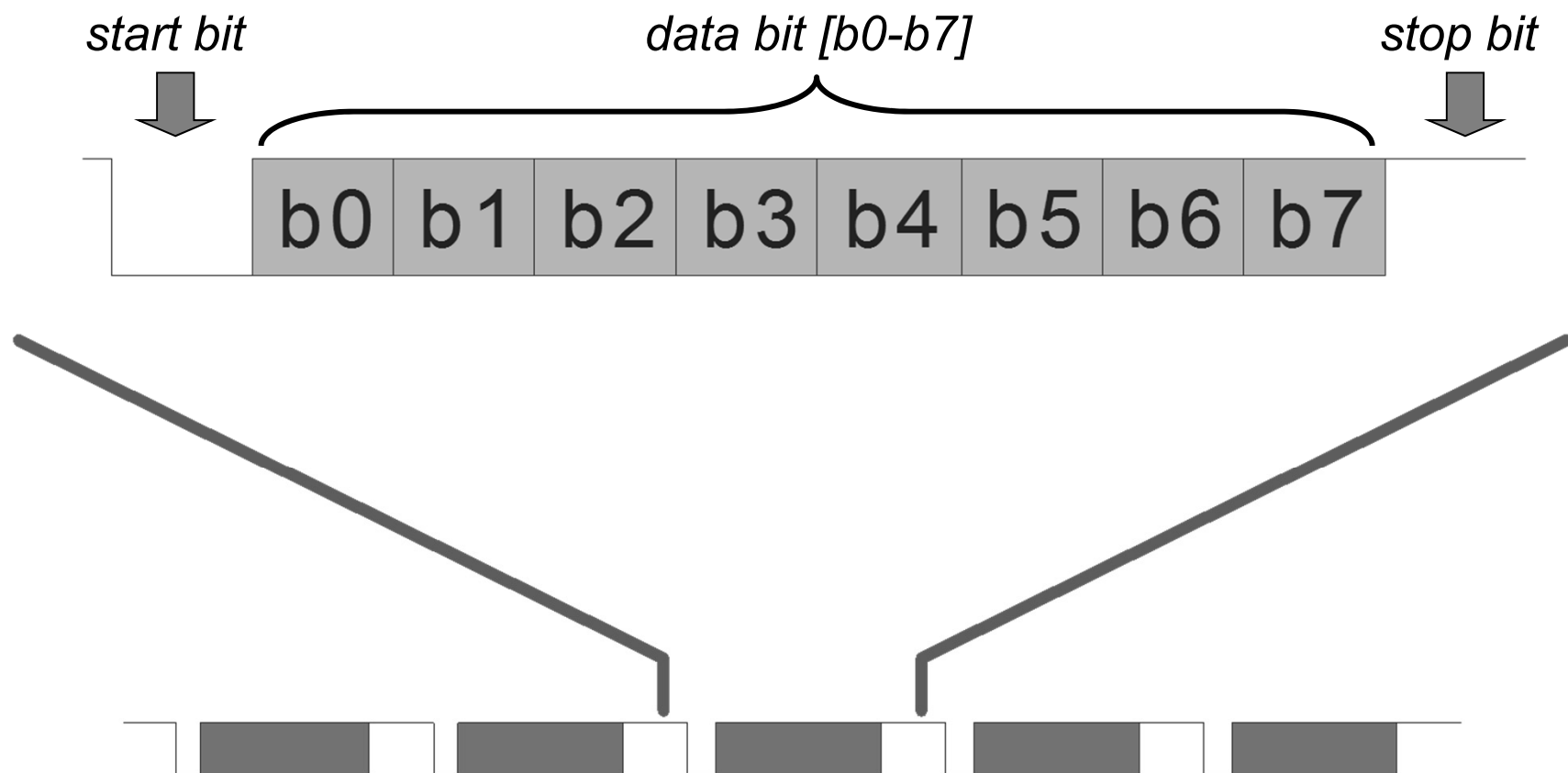
# Comunicação série - 8051

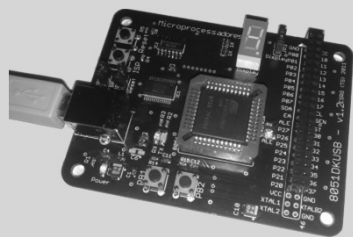
- **Interface série no microcontrolador 8051**
  - **Modo 1:**
    - *São enviados 10 bits de cada vez: 1 start bit, 8 bits de dados (LSB é enviado primeiro) e 1 stop bit*
    - *Na recepção o stop bit é guardado em RB8 que faz parte do registo SCON*
    - *O baud rate é variável*



# Comunicação série - 8051

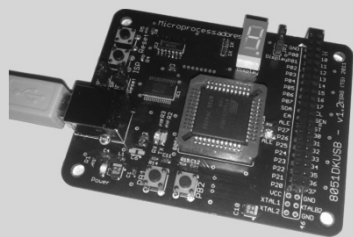
- Interface série no microcontrolador 8051





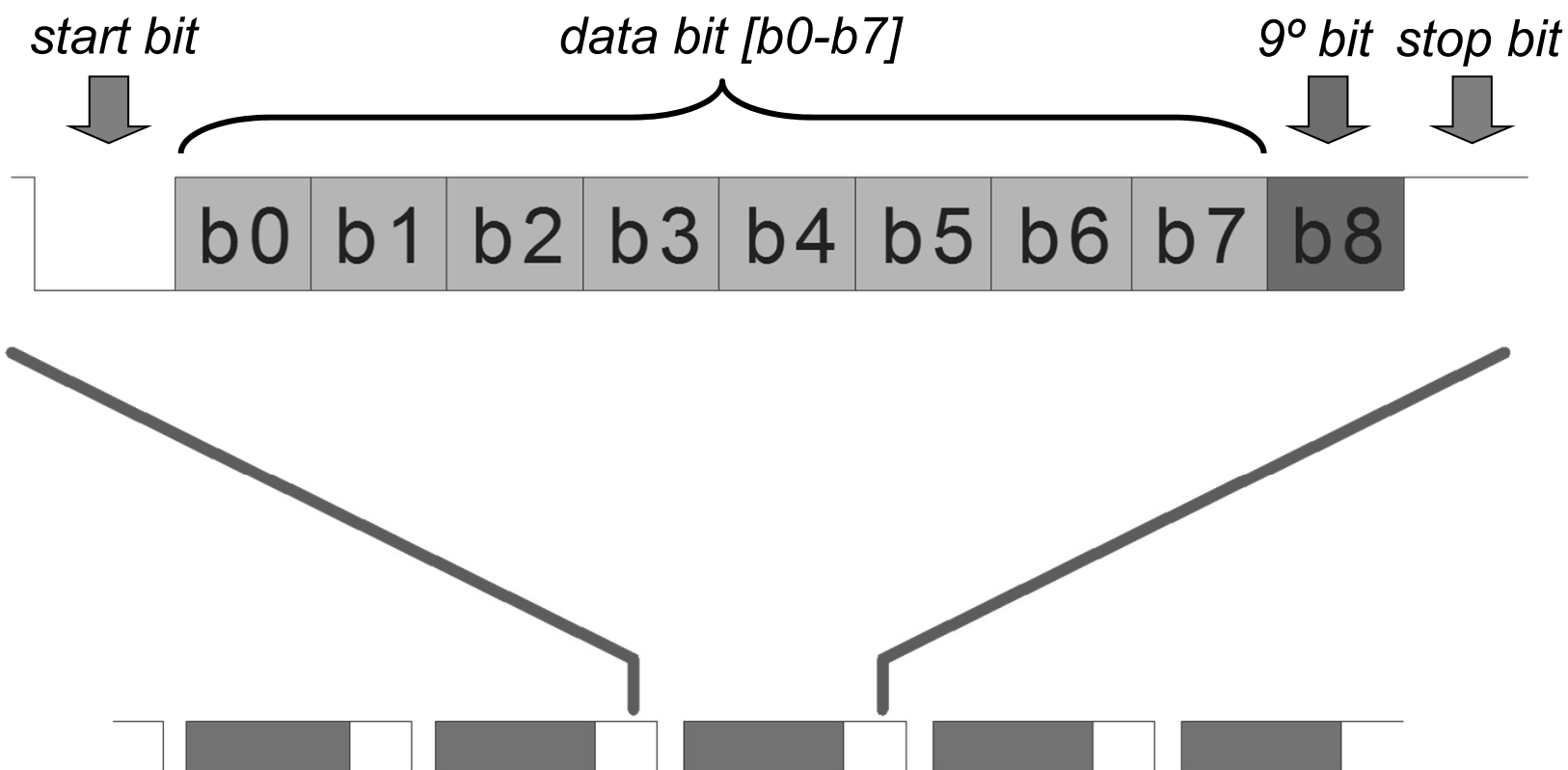
# Comunicação série - 8051

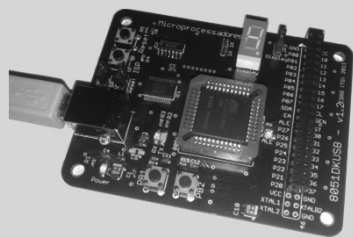
- **Interface série no microcontrolador 8051**
  - **Modo 2:**
    - São transmitidos (através de TxD) ou recebidos (através de RxD) 11 bits: 1 start bit, 8 bits de dados (LSB é enviado primeiro), 1 nono bit (cujo valor é programável – TB8) e 1 stop bit
    - Na recepção o nono bit é guardado em RB8
      - Nota: O nono bit pode ser usado como bit de paridade para correcção de erros ou na implementação de protocolos especiais para ligação de múltiplos processadores
    - O baud rate pode ser programado como 1/32 ou como 1/64 da frequência do oscilador



# Comunicação série - 8051

- Interface série no microcontrolador 8051





# Comunicação série - 8051

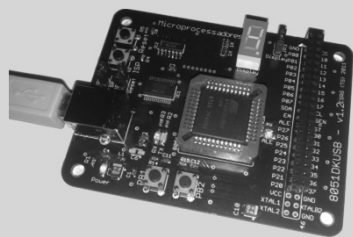
- **Interface série no microcontrolador 8051**

- ***Modo 3:***

- *São transmitidos (através de TxD) ou recebidos (através de RxD) 11 bits: 1 start bit, 8 bits de dados (LSB é enviado primeiro), 1 nono bit (cujo valor é programável – TB8) e 1 stop bit*
- *Na recepção o nono bit é guardado em RB8*
- *O baud rate é variável*

*Nota:*

*Este modo de operação é semelhante ao modo 2, excepto que o baud rate é variável*



# Comunicação série - 8051

- **Interface série no microcontrolador 8051**

- *Cálculo do baud rate:*

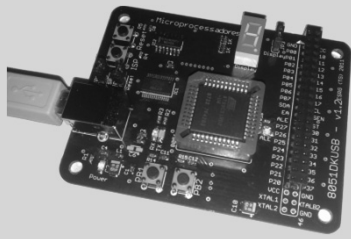
- *No modo 0, o baud rate é dado por*

$$baud_{rate} = \frac{f_{oscillator}}{12}$$

- *No modo 2, o baud rate depende do bit SMOD do registo PCON, sendo dado por*

$$baud_{rate} = \frac{2^{SMOD}}{64} f_{oscillator}$$





# Comunicação série - 8051

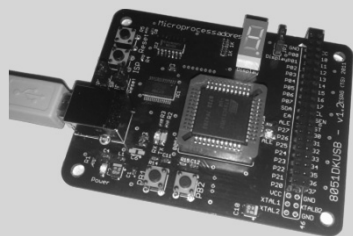
- *Cálculo do baud rate (usando o timer 1):*

- *Nos modos 1 e 3, o baud rate é dado por*

$$baud_{rate} = \frac{2^{SMOD}}{32} f_{(Timer1\ overflow\ rate)}$$

- *Quando o timer 1 é programado no seu modo de auto-reload, o baud rate será dado por*

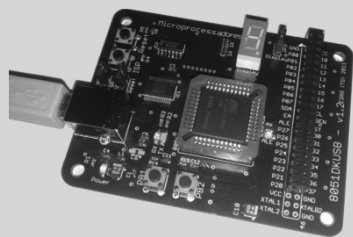
$$baud_{rate} = \frac{2^{SMOD}}{32} \frac{f_{oscillator}}{12 \times [256 - (TH_1)]}$$



# Comunicação série - 8051

- Calculando o baud rate (usando o timer 1):

Baud Rate	f <sub>osc</sub>	SMOD	Timer 1		
			C/T	Mode	Reload Value
Mode 0 Max: 1 MHz	12 MHz	X	X	X	X
Mode 2 Max: 375K	12 MHz	1	X	X	X
Modes 1, 3: 62.5K	12 MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEDH



# Comunicação série - 8051

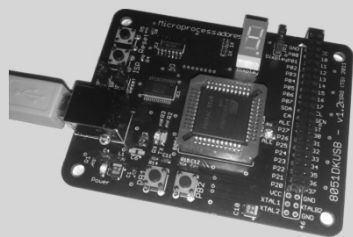
- *Cálculo do baud rate (usando o timer 2):*

- *Nos modos 1 e 3, o baud rate é dado por:*

$$baud_{rate} = \frac{f_{(Timer\ 2\ overflow\ rate)}}{16}$$

- *Quando o timer 2 é programado no seu modo de auto-reload, o baud rate será dado por:*

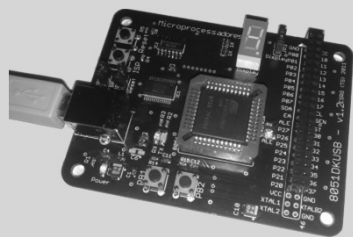
$$baud_{rate} = \frac{f_{oscillator}}{32 \times [65536 - (RCAP2H, RCAP2L)]}$$



# Comunicação série - 8051

- *Baud rate (usando o timer 2):*
  - *Timer 2 em modo de gerador de baud rate;*
  - *Interface série em modo 1 ou 3;*
  - *Ver páginas 14-18 do datasheet do 89C51Rx2.*

Baud Rate		Osc Freq	Timer 2	
12 clock mode	6 clock mode		RCAP2H	RCAP2L
375 k	750 k	12 MHz	FF	FF
9.6 k	19.2 k	12 MHz	FF	D9
2.8 k	5.6 k	12 MHz	FF	B2
2.4 k	4.8 k	12 MHz	FF	64
1.2 k	2.4 k	12 MHz	FE	C8
300	600	12 MHz	FB	1E
110	220	12 MHz	F2	AF
300	600	6 MHz	FD	8F
110	220	6 MHz	F9	57



# Comunicação série - 8051

- *Exemplo:*
  - *Programação da comunicação série, usando a UART no modo 1: 8 bits de dados, mais start bit e stop bit. O baud rate deve ser 19200 bps*

*; programação do timer: TMOD, TH0, TL0 e TCON*

*mov TMOD, #20H ; Timer 1 no modo 2*

*mov TH1, #0FDH ; baud rate= 19200 bps*

*mov TL1, #0FDH*

*setb TR1*

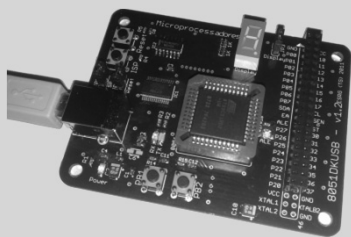
*mov A, PCON ; duplicar o baudrate,*

*setb ACC.7 ; colocando o bit SMOD*

*mov PCON, A ; a 1*

*; programação de SCON*

*mov SCON, #0B0H ; Modo 1, SM2= 1, REN= 1*



# Comunicação série - 8051

- Interface série no microcontrolador 8051

- *Exemplo*

*; aplicação: Fazer o echo do caracter recebido*

```
loop: jnb     RI, loop
      clr     RI
      mov     A, SBUF
      mov     SBUF, A
      jnb     TI, $
      clr     TI
      jmp     loop

      end
```