

Microprocessadores

COMUNICAÇÃO SERIAL

Programando Serial

1	P _{1.0}	V _{CC}	40
2	P _{1.1}	ED ₀ P _{0.0}	39
3	P _{1.2}	ED ₁ P _{0.1}	38
4	P _{1.3}	ED ₂ P _{0.2}	37
5	P _{1.4}	ED ₃ P _{0.3}	36
6	P _{1.5}	ED ₄ P _{0.4}	35
7	P _{1.6}	ED ₅ P _{0.5}	34
8	P _{1.7}	ED ₆ P _{0.6}	33
9	RST	ED ₇ P _{0.7}	32
10	P _{3.0} RXD	EA/V _{PP}	31
11	P _{3.1} TXD	ALE/ $\overline{\text{PROG}}$	30
12	P _{3.2} $\overline{\text{INT0}}$	$\overline{\text{PSEN}}$	29
13	P _{3.3} $\overline{\text{INT1}}$	E ₁₅ P _{2.7}	28
14	P _{3.4} T ₀	E ₁₄ P _{2.6}	27
15	P _{3.5} T ₁	E ₁₃ P _{2.5}	26
16	P _{3.6} $\overline{\text{WR}}$	E ₁₂ P _{2.4}	25
17	P _{3.7} $\overline{\text{RD}}$	E ₁₁ P _{2.3}	24
18	X _{TAL2}	E ₁₀ P _{2.2}	23
19	X _{TAL1}	E ₉ P _{2.1}	22
20	V _{SS}	E ₈ P _{2.0}	21

8051

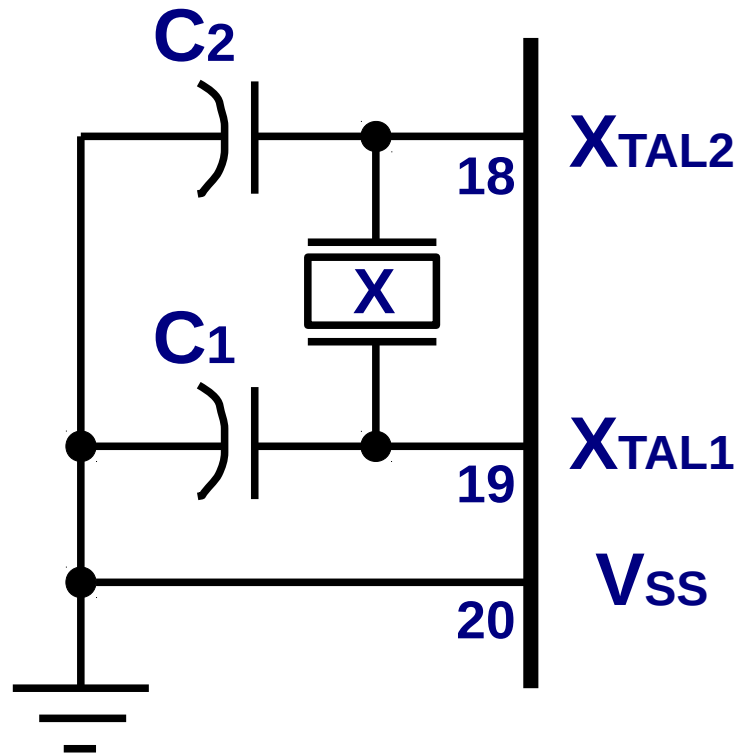
Programando Serial

As *baud rate's* para transmissão serial são: 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 e 115200 bit/s. Para se atingir essas taxas é necessário realizar a configuração do hardware de modo que possa receber a programação.

Descrição Técnica

X_{TAL1} | X_{TAL2}

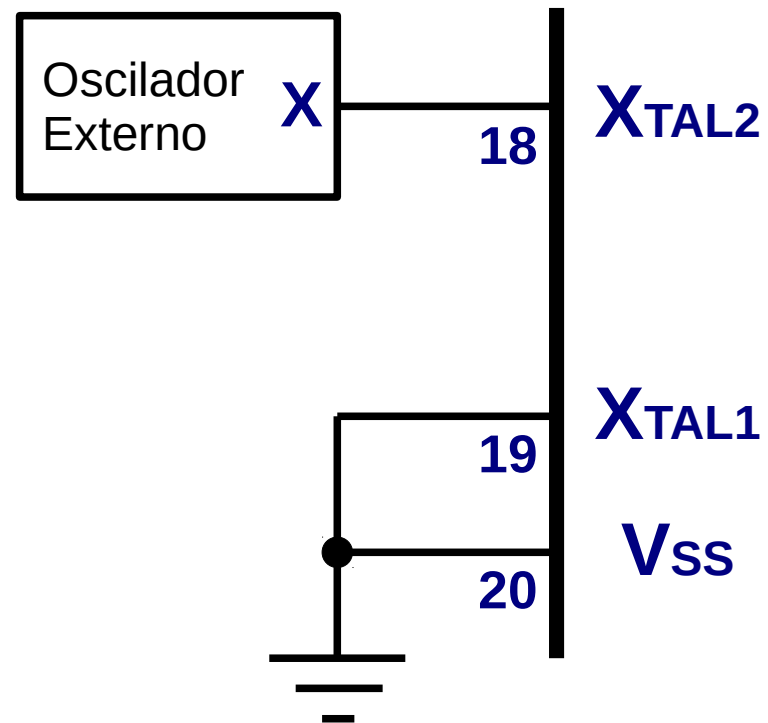
- 3MHz a 12MHz



Descrição Técnica

X_{TAL1} | X_{TAL2}

- 3MHz a 12MHz



Programando Serial

Baud Rate:

Considerando-se XTAL = 11,0592MHz no Modo 1, encontrar o valor necessário do *Timer* para ter as seguintes *baud rate's*:

- (a) 9600 bit/s
- (b) 4800 bit/s
- (c) 2400 bit/s
- (d) 1200 bit/s

Programando Serial

Baud Rate:

Para o Modo 1 temos: $SM0 = 0$ | $SM1 = 1$ | $SM2 = 0$.

$XTAL = 11,0592\text{MHz}$, temos:

$\text{Clock interno} = 11,0592 \text{ MHz} \div 12 = 921,6 \text{ kHz}$

$\text{UART} = 921,6 \text{ kHz} / 32 = 28800\text{Hz}$

Programando Serial

Baud Rate:

$$(a) \rightarrow \frac{28800}{3} = 9600 \text{ bps} \gg \text{FDh}$$

$$(b) \rightarrow \frac{28800}{6} = 4800 \text{ bps} \gg \text{FAh}$$

$$(c) \rightarrow \frac{28800}{12} = 2400 \text{ bps} \gg \text{F 4 h}$$

$$(d) \rightarrow \frac{28800}{24} = 1200 \text{ bps} \gg \text{E 8 h}$$

Programando Serial

<i>Baud rate (bps)</i>	<i>Clock (MHz)</i>	<i>SMOD</i>	<i>Timer1</i>		
			<i>CT</i>	<i>Modo</i>	<i>Reload</i>
1 M	12	X	X	X	X
375k	12	1	X	X	X
62,5k	12	1	0	2	FFh
19,2k	11,059	1	0	2	FDh
9,6k	11,059	0	0	2	FDh
4,8k	11,059	0	0	2	FAh
2,4k	11,059	0	0	2	F4h
1,2k	11,059	0	0	2	E8h
137,5	11,059	0	0	2	1Dh
110	6	0	0	2	72h
110	12	0	0	1	FFEBh

Programando Serial

Exercício 1:

Escrever um programa para transferir a letra “A” serialmente a uma taxa de 4800 baud, continuamente.

Programando Serial

Exercício 1:

Escrever um programa para transferir a letra “A” serialmente a uma taxa de 4800 baud, continuamente.

Programando Serial

Exercício 1:

Serial: Modo 1 [SCON = 40h]

Multiprocessador: Não [SM2 = 0]

Timer1: Modo 2 [TMOD = 20h]

Baud Rate = 4800baud [TH1 = FAh | SMOD = 0]

SM0	SM1	SM2	REN	TE8	RB8	TI	RI
-----	-----	-----	----------------	----------------	----------------	----	---------------

Programando Serial

A importância do flag TI

1. O caracter (byte) para ser transferido deve ser escrito no registrador SBUF (99h).
2. Transfere o start bit
3. O caracter é transmitido bit a bit (LSB → MSB).
4. O stop bit é transferido.

Programando Serial

A importância do flag TI

5. Durante a transferência do stop bit o 8051 faz $TI = 1$.
6. Monitorando o TI evitamos perder dados.
7. Após o SBUF é carregado com o novo caracter (byte).
8. Em seguida, o TI deve ser limpo para que o novo caracter seja transferido.

Programando Serial

Exercício 2:

Escrever um programa para receber bytes serialmente e colocá-los em P1, continuamente. Setar a taxa de transmissão para 4800 baud.

Programando Serial

Exercício 2:

Serial: Modo 1 [SCON = 50h]

Multiprocessador: Não [SM2 = 0]

Timer1: Modo 2 [TMOD = 20h]

Baud Rate = 4800baud [TH1 = FAh | SMOD = 0]

SM0	SM1	SM2	REN	TE8	RB8	TI	RI
-----	-----	-----	-----	----------------	----------------	---------------	----

Assembly

The diagram illustrates an assembly program with two labels, *INICIO* and *RX*, and a control flow loop. Two vertical arrows on the left side indicate the flow of execution: one pointing downwards from the *INICIO* label to the *RX* label, and another pointing upwards from the *RX* label back to the *INICIO* label, forming a loop. The code is as follows:

```
ORG 0000h
LJMP INICIO
ORG 0040h
INICIO: MOV TMOD, #20h
        MOV TH1, #0FAh
        MOV SCON, #50h
        SETB TR1
RX:     JNB RI, RX
        MOV A, SBUF
        MOV P1, A
        CLR RI
        SJMP RX
END
```

Programando Serial

Exercício 3:

Escrever um programa para transferir a palavra “CASA” serialmente a uma taxa de 110 bauds uma única vez e parar o programa em uma linha fixa.

Referências...

NICOLOSI, P. **Microcontrolador 8051**: detalhado. São Paulo: Érica, 2010.

BALBINOT, A. **O Microcontrolador 8051**. Notas de aula. São Paulo: 2010.

GIMENEZ, S.P. **Microcontrolador 8051**. São Paulo: Pearson Education do Brasil, 2005.

