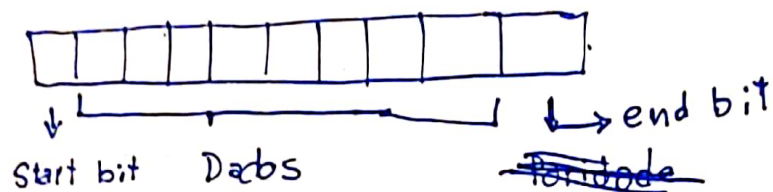


1-a) Modo 0: - 8 bits de transmissão

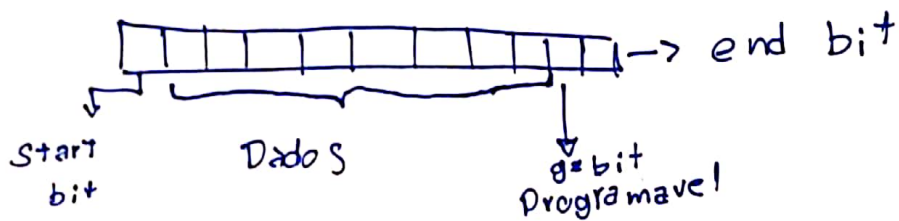
- baud rate fixo =  $1/12 \cdot \text{freq.}$
- Transmite e recebe por RXD

Modo 1:  $\rightarrow$  10 bits transmitidos por TXD e recebidos por RXD

- Baud rate variável ( $T_1$ )



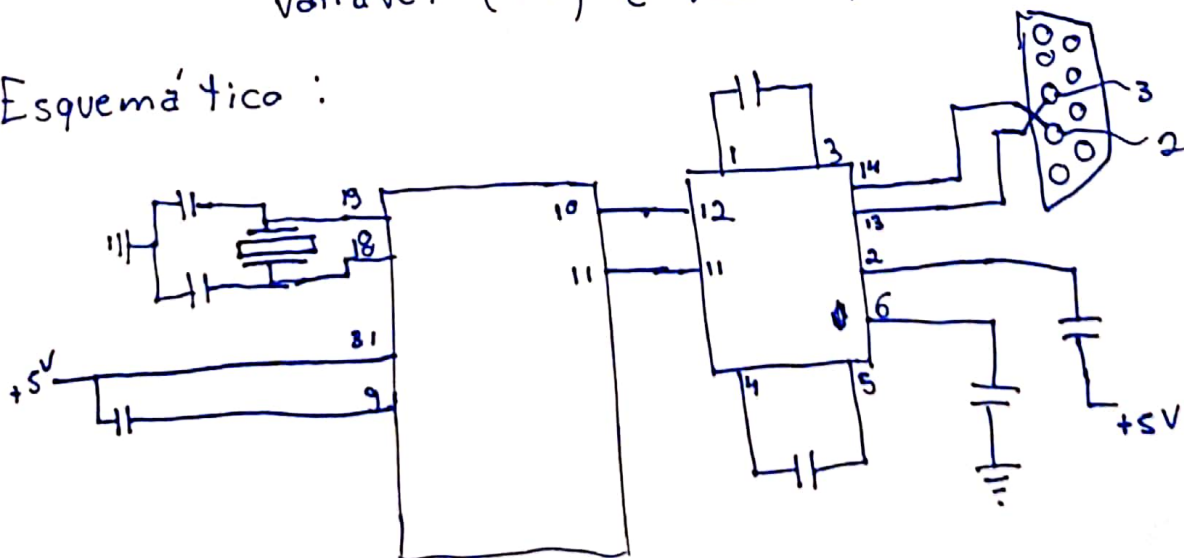
Modo 2: - 11 bits com TXD e RXD



- Baud rate ~~vá~~ usa  $T_1$  e S.MOD (PCON.7)

Modo 3: - Igual ao modo 2, porém baud rate é variável ( $T_1$ ) e não depende de S.MOD

Esquemático:



1- b) SMO → Modo de comunicação Serial 1

SM1 → Modo de comunicação Serial 1

SM2 → Multiprocessamento (é outro 8051)

REN → Habilita recepção pelo pino P3.0/RXD

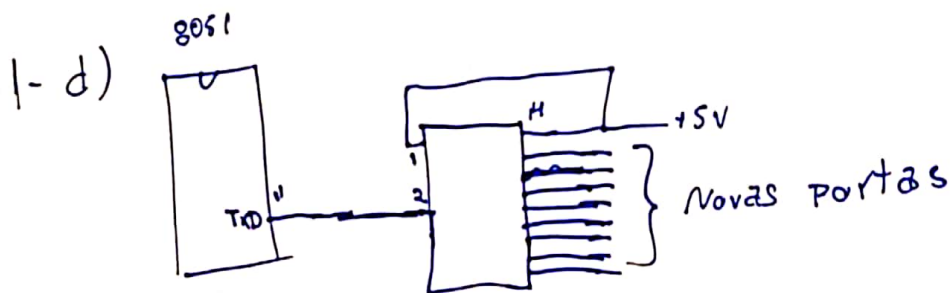
TBB → Bit extra a ser transmitido (Modos 2/3)

RBB → Bit extra a ser recebido (Modos 2/3)

TI → Flag de interrupção no stop bit / 8º bit (modo 0)

RI → Flag de interrupção de recebimento

1- c) A transmissão ocorre em paralelo utilizando as portas dos microcontroladores para transmitir os dados.



Usando R0 como uma nova saída "P4":

loop:

mov A, @R0

~~jmp ter~~

jz terminou

mov SBUF, A

jnb TI, \$

cir TI

; Faz algo, rotina normal

jmp loop

~~terminou:~~

nop

$$2 - \text{baud} = \frac{62500}{256 - \text{TH1}}$$

TH1 = #FDh

clr smo

setb sm1 ; Modo de op. 1

setb ren ; habilita recepção

Configurando SCON

mov A, PCON

clr ACC.7 ; SMOD = 0

mov PCON, A

Configurando SMOD

mov TMOD, #00100000b ; Modo 2

mov TH1, #0FDh

mov TL1, #0FDh

setb TR1 ; Inicia timer

Configurando T1

mov RO, #40h ; pos inicial de memória

mov B, #09h ; n. de bytes a ser lido

loop:

jnb RI, \$

clr RI

mov A, SBUF

cjne A, #AAh, loop

jmp found AAH loop

jmp loop

found AAH loop:

jnb RI, \$

clr RI

mov A, SBUF

djnz B, prox

mov RO, #40h

mov B, #09h

clr P1.0  
jmp loop

prox:

mov @RO, A

inc RO

jmp found AAH loop

Lendo os 8 bits após #AAh

Procurando por #AAh

3-  $\text{baudrate} = 2400$   
 $\text{SMOD} = 0$   
 $\text{Xtal} = 12 \text{ MHz}$  }  $T_1 = \#0F3h$

org 0000h  
 jmp main

org 0023h  
 call receive  
 reti

org 003h

main:

mov SP, #04fh  
 mov PSW, #00h  
 mov IE, #90h  
 clr SMO  
 setb SM1  
 setb REN  
 mov A, PCON  
 setb clr ACC.7  
 mov pcon, A  
 mov TMOD, #00100000b  
 mov TH1, #0F3h  
 mov TL1, #0F3h  
 mov 20h, #'?'

loop:  
 jmp \$

receive:

clr RI  
 mov A, SBUF  
 mov C, D  
 mov FO, C  
 jnb ACC.7, naoC  
 jnb FO, errado  
 jmp ~~errado~~ correto

naoC:  
 jnb FO, correto  
 jmp errado

correto:

clr PI.0  
 jmp fim

errado:

jmp transmitir

ret

transmitir:

mov A, 20h  
 mov SBUF, A  
 jnb, TI, \$  
 clr TI  
 jmp fim

fim:

nop