

## NOTAS DE AULAS: MICROCONTROLADORES

### Comunicação Serial

**Prof. João Perea Martins**  
**Dep. De Computação, FC-UNESP**  
**E-mail: joao.perea@unesp.br**

Registadores utilizados na programação da comunicação serial do PIC.

#### 1) Registradores Principais:

TXSTA Transmit Status and Control Register

RCSTA Receive Status and Control Register

SPBRG Baud Rate Generator

#### 2) Buffers

TXREG Buffer de transmissão

RCREG Buffer de recepção (2 bytes)

#### 3) Registradores Auxiliares:

PIE1 Peripheral Interrupts Enable

PIR1 Individual flag bits for the peripheral interrupt

#### 1) TXSTA: Transmit Status And Control Register

##### TXSTA

7	6	5	4	3	2	1	0
CSRC	TX9	TXEN	SYNC	X	BRGH	TRMT	TX9D

CSRC: Fonte do clock

Modo assíncrono: não importa.

TX9: habilita transmissão de 9º bits

1 = Selecciona a transmissão de 9 bits

0 = Selecciona a transmissão de 8 bits

TXEN: Habilita transmissão

1 = transmissão ativada

0 = Transmitir desativado

SYNC: Seleção do Modo de Operação

1 = modo síncrono

0 = modo assíncrono

BRGH: Seleção de taxa para cálculo da baud rate (a escolha vai depender do cálculo).

Só é usado no modo Assíncrono

1 = alta velocidade

0 = velocidade baixa

TRMT: Status do registrador de transmissão.

1 = TSR vazio

0 = TSR cheio

TX9D: 9º bit de dados de transmissão.

Pode ser bit de paridade

## 2) RCSTA: Receive Status And Control Register

### RCSTA

7	6	5	4	3	2	1	0
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D

SPEN: Habilitação da porta serial

1 = Porta serial ativada

(Os pinos RB1/RX e RB2/TX para serial. TRISB <2:1> deve ser setado).

0 = porta serial desativada

RX9: habilitação de recebimento de 9 bits

1 = Seleciona a recepção de 9 bits

0 = Seleciona a recepção de 8 bits

SREN: Bit de habilitação de recebimento único

Modo assíncrono: não importa.

CREN: Habilita recebimento contínuo

Modo assíncrono: 1 = Permite recebimento contínuo

0 = desativa o recebimento contínuo

ADDEN: Habilitação de detecção de endereço

Modo síncrono: Não é usado

Modo assíncrono: 1 = Habilita a detecção de endereços

0 = Desativa a detecção de endereço

FERR: bit de erro de enquadramento. Indica se se houve problema na detecção do Stop-Bit

1 = Erro de enquadramento (Framig)

0 = sem erro de enquadramento

OERR: bit de erro de Overrun (O buffer já está cheio e chegam novos bytes)

1 = Erro de overrun

0 = sem erro

RX9D: 9º bit de dados

9º bit de dados recebidos (pode ser o valor do bit de paridade recebido)

## 3) SPBRG: USART Baud Rate Generator

O SPBRG determina a taxa de transmissão e depende do bit: TXSTA, BRGH

Se BRGH=0:

$$SPBRGH = \left( \frac{F_{osc}}{64 \cdot bps} \right) - 1 \quad (1)$$

Se BRGH=1:

$$SPBRGH = \left( \frac{F_{osc}}{16 \cdot bps} \right) - 1 \quad (2)$$

Valores usuais de bps: 1200, 2400, 4800, 9600, 19200, 38400, 57600 e 115200 bps

## 4) Buffers de comunicação: TXREG e RCREG

**TXREG:** Funciona como um Buffer de Transmissão

**RCREG:** Funciona como um Buffer de Recepção

## 6) PIE1: Peripheral Interrupts Enable

### PIE1

7	6	5	4	3	2	1	0
		RCIE	TXIE				

PIE,RCIE = 0      Desabilita interrupção de Recepção  
PIE,RCIE = 1      Habilita interrupção de Recepção

PIE,TXIE = 0      Desabilita interrupção de Transmissão  
PIE,TXIE = 1      Habilita interrupção de Transmissão

## 7) PIR1: individual flag bits for the peripheral interrupt

### PIR1:

7	6	5	4	3	2	1	0
		RCIF	TXIF				

PIE,RCIF = 0      Não houve interrupção de Recepção  
PIE,RCIF = 1      Houve interrupção de Recepção

PIE,TXIF = 0      Não houve interrupção de Transmissão  
PIE,TXIF = 1      Houve interrupção de Transmissão

## ALGORITMOS BÁSICOS:

### 1) Programação:

```
TRISB = b 'xxxx x11x'  
TXSTA = b '0010 x110'  
RCSTA = b '1001 0000 '  
SPBRS = valor-calculado
```

### 2) Lógica de recepção de um byte.

```
LOOP: BTFSS PIR1,RCIF      ; Chegou algum byte pela serial ?  
      GOTO LOOP            ; Não  
      MOVF RCREG,W        ; Sim, então tiro o byte do buffer de recepção
```

O comando RCREG,W zera automaticamente o bit PIR1,RCIF

### 3) Lógica de transmissão de um byte.

```
WAIT: BTFSS TXSTA,TRMT    ; O buffer de transmissão está vazio ?  
      GOTO WAIT            ; Não  
      MOWF TXREG          ; Sim, então coloco dado no buffer para ser enviado
```

Para a transmissão, o registrador TXREG descarrega o byte em outro mecanismo de hardware chamado TSR e p bit PIR1,TXIF é setado. Quando o byte é completamente enviado pelo hardware o byte TXSTA, TRMT é setado.