

# **Manual de utilização**

# **Do**

# **Remoto Serial**

(a partir da versão 1.27 Beta)

Rev. 0 em 24.09.2003

Copyright Junho/2003 Renie S. Marquet



## Índice:

Notas sobre o manual e os softwares .....	5
O que é o Remoto Serial .....	6
<b>Componentes do Remoto Serial PC</b>	
Áreas da tela principal .....	7
Menu principal .....	8
Menu Dados .....	8
Menu Editor .....	8
Menu Dispositivo .....	9
Menu Execução .....	9
Menu Outros .....	10
Menu Ajuda .....	10
Barra de Ferramentas .....	11
Conversor de códigos .....	12
Editor de Origem .....	13
Área de bytes enviados .....	14
Área de bytes recebidos .....	14
Barra de status .....	15
<b>Configurações</b>	
Configurações do dispositivo serial .....	16
Configurações do editor .....	17
Configurações da execução .....	18
Comandos do Remoto Serial PC .....	19
Macros do Remoto Serial PC .....	20
Comandos do Remoto Serial PIC .....	26
Possíveis implementações futuras .....	27
<b>Esquema eletrônico para comunicação</b>	
entre Remoto Serial PC e Remoto Serial PIC .....	28
Exemplos .....	29



## **Notas sobre o manual e os softwares:**

**Este manual foi escrito tendo como base a versão 1.27 Beta do Remoto Serial PC e a versão 1.2 do Remoto Serial PIC. As implementações incluídas em versões posteriores serão indicadas próximo das instruções com o número da versão entre colchetes, exemplos:**

**[ 1.29 PC] - implementado a partir da versão 1.29 do Remoto Serial PC.**

**[ 2.10 PIC] - implementado a partir da versão 2.10 do Remoto Serial PIC.**

**O texto contendo dados/comandos que serão interpretados pelo Remoto Serial PC para serem enviados ao Remoto Serial PIC é considerado a origem dos dados. Deste modo, o texto foi batizado de ORIGEM e assim será tratado tanto pelo software como neste manual.**

**Como ambos os softwares e o manual foram desenvolvidos, testados e depurados somente pelo autor, em pequenos lapsos de tempo que o mesmo dispõe, estes poderão apresentar algum ou alguns bugs e erros.**

**Qualquer comunicado de bug, erro ou sugestões serão bem recebidos pelo autor.**

**Nenhuma garantia é oferecida com os softwares/manual, o uso dos mesmos é de responsabilidade única dos usuários, não cabendo ao autor nenhum encargo advindo de sua utilização, quer direta ou indiretamente.**

**Os softwares e outros arquivos referentes ao Remoto Serial podem ser baixados do site do autor. Como a Internet não é estática, o site do autor pode ser realocado em outro servidor, como já aconteceu. Para manter-se informado de novas versões, mudanças de endereço, etc. envie um e-mail ao autor demonstrando seu interesse.**

**Estes softwares são livres para uso não comercial, qualquer utilização diferente desta, bem como a distribuição em CDs, ou outros meios que não o site do autor, deverá ser solicitada, mediante autorização prévia ao autor, incorrendo em violação de direitos autorais o desrespeito a essas regras.**

## **O que é o Remoto Serial:**

O Remoto Serial é um aplicativo originalmente criado para comunicação assíncrona entre um computador PC e um microcontrolador PIC16F877A (® Microchip) utilizando somente as linhas RX e TX, o que não significa que sua utilização esteja restrita a esse fim.

O aplicativo é composto de dois programas distintos, um executando no PC (Remoto Serial PC) desenvolvido em Delphi 5.0 (® Borland) e outro desenvolvido em assembly MPASM (® Microchip) executando no PIC (Remoto Serial PIC) .

O programa que roda no PC é responsável por interpretar os dados apresentados no formato de texto a serem enviados ao PIC; o mesmo apresenta uma grande gama de facilidades, inclusive macros (veja: Comandos do Remoto Serial PC e Macros do Remoto Serial PC).

O Remoto Serial PIC - Um PIC16F877A escravo, rodando um pequeno programa , que conforme o "comando" recebido pela serial (comunicação assíncrona), altera qualquer registro interno, ou os lê, devolvendo seu conteúdo pela serial. Deste modo, é possível fazer o PIC interagir com o "mundo exterior", configurando dinamicamente PORTs, definir ou ler os dados presentes nos pinos dos PORTs, configurar AD, ler valores do AD, etc. etc.

## Apresentação da tela principal:

The screenshot shows the main interface of the 'Remoto Serial v1.2 Pré' application. The window title is 'Remoto Serial v1.2 Pré - Origem: C:\Users\Renie...neus\remoto\LCD40X2\_8bori.ORI'. The interface includes a menu bar (Dados, Editor, Dispositivo, Execução, Outros, Ajuda), a toolbar with icons for file operations and execution, and a data input section with fields for Hexa (41), Dec (65), Binário (01000001), and ASCII. Below this is a 'Base default' section with radio buttons for Decimal (selected), Hexa, Octa, and Binário. The main area is a text editor containing configuration code for an LCD. At the bottom, there are two data display tables for sent and received bytes, and a status bar showing 'Serviço aberto', 'Env. 1517', 'Rec. 0', and '0:00:01'. Callouts identify the 'Menu principal', 'Barra de ferramentas', 'Editor de texto dos dados de Origem', 'Conversor', 'Base default', 'Área de visualização dos bytes efetivamente enviados', 'Área de visualização dos bytes recebidos', 'Barra de redimensionamento das áreas', and 'Barra de Status'.

Menu principal

Barra de ferramentas

Conversor

Base default

Barra de redimensionamento das áreas

Barra de Status

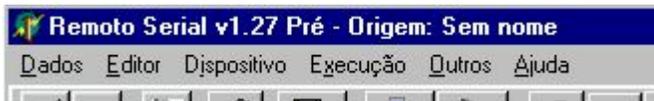
Área de visualização dos bytes efetivamente enviados

Área de visualização dos bytes recebidos

Editor de texto dos dados de Origem

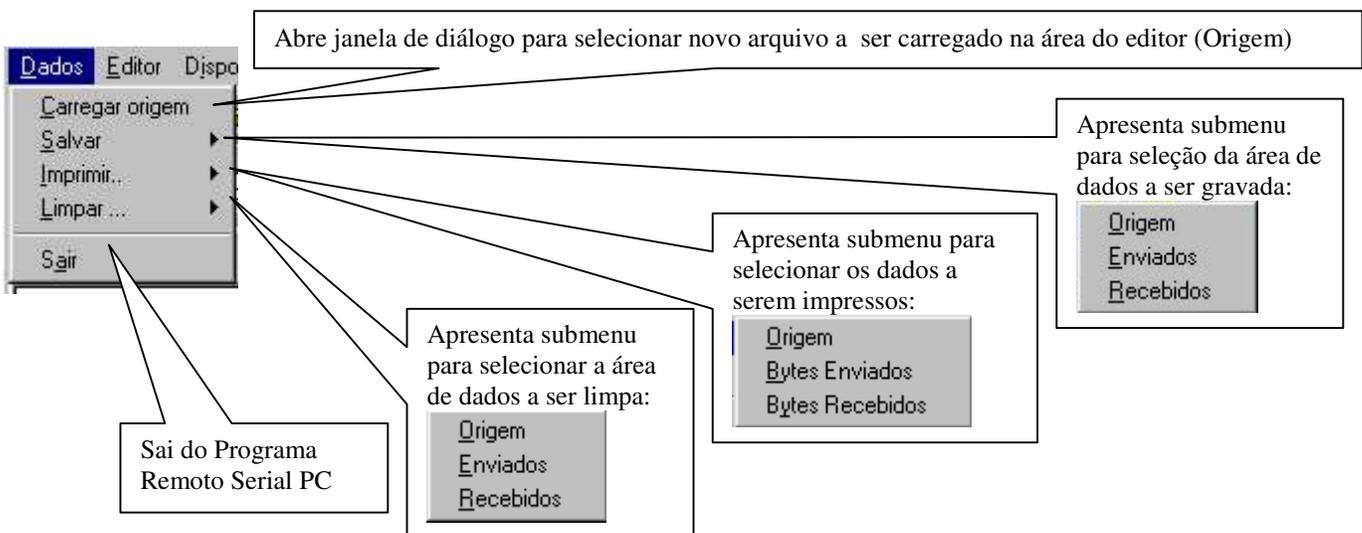
Enviados (hexa)	Caracter	Recebidos (hexa)	Caracter
31 53 01 06 00 53 01 08	; 1S...S..		
00 53 00 06 38 53 00 08	; .S..8S..		
80 53 00 08 00 53 00 08	; €S...S..		
80 53 00 08 00 53 00 06	; €S...S..		
06 53 00 08 80 53 00 08	; .S...€S..		
00 53 00 06 08 53 00 08	; .S...S..		

## Menu principal:



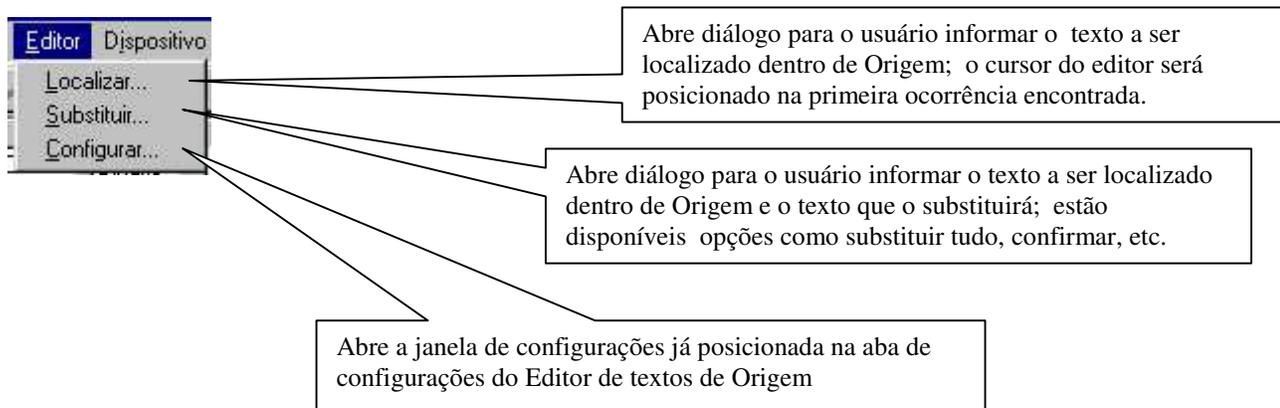
O Menu principal com suas opções e submenus apresentam todos os comandos disponíveis no ambiente de trabalho do Remoto Serial PIC, assim como algumas opções de configuração das áreas visíveis na tela principal etc. Algumas opções relacionadas a comunicação com o PIC só estarão disponíveis após iniciar a comunicação, outras, relativas a execução só estarão disponíveis durante o envio de dados pela serial.

## Menu Dados:



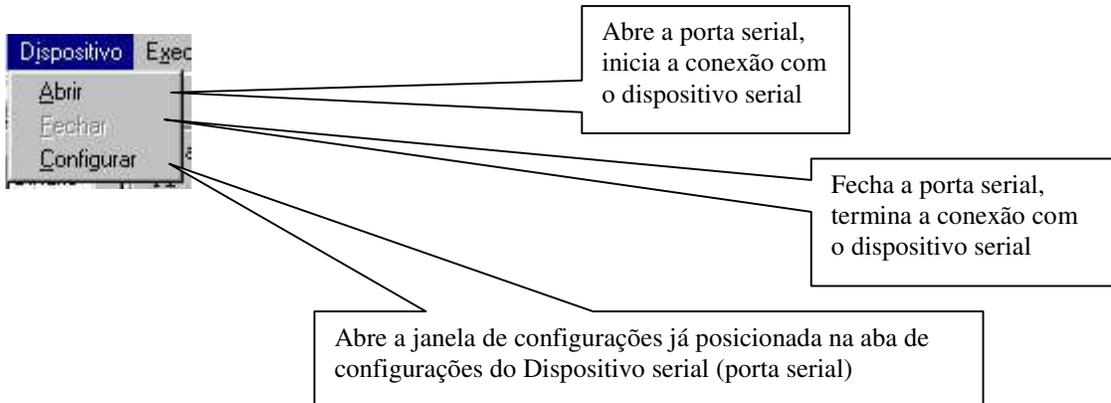
## Menu Editor:

O menu editor apresenta opções para se trabalhar com o texto de Origem

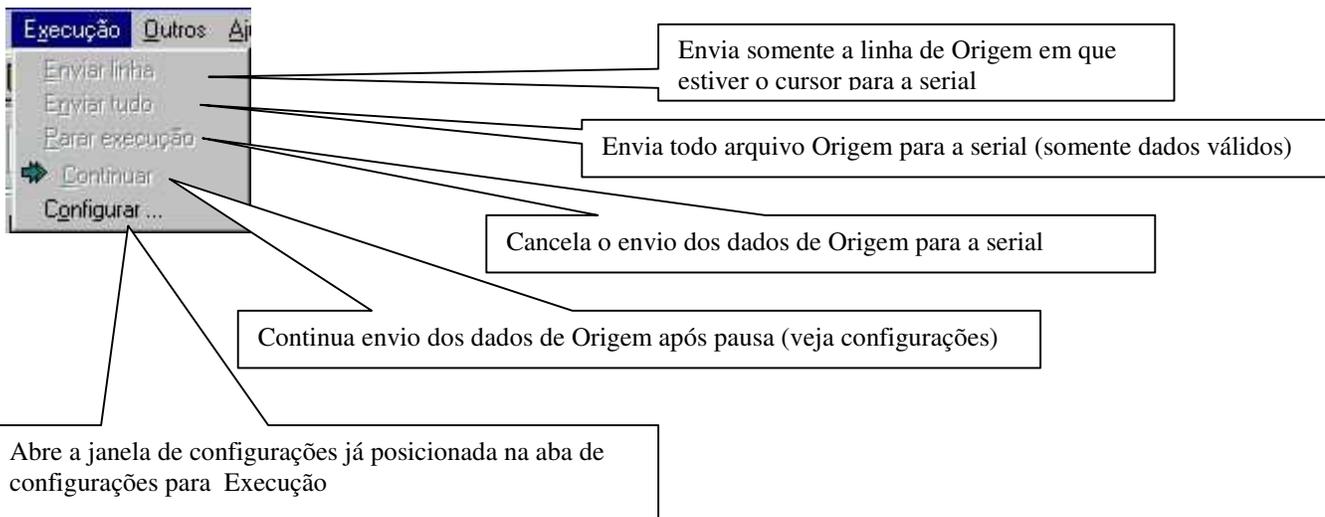


## Menu Dispositivo:

O menu dispositivo apresenta as opções relativas ao dispositivo serial.

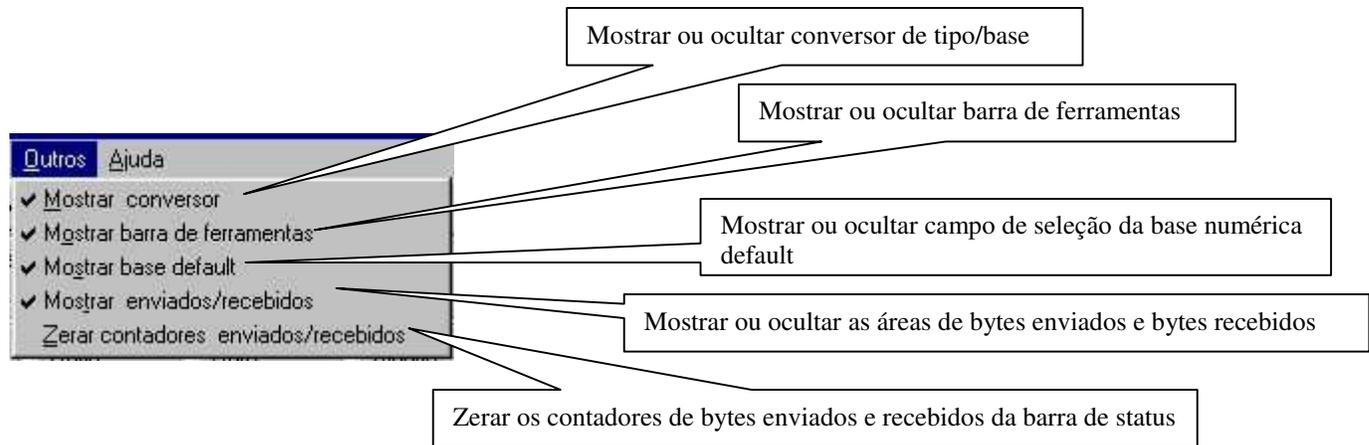


## Menu Execução:

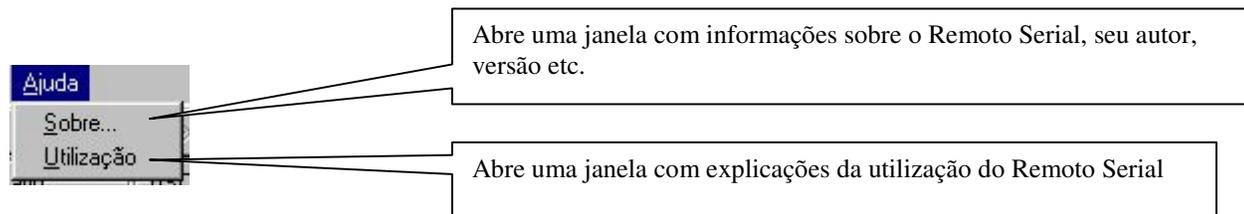


## Menu Outros:

O menu outros apresenta diversas opções relativas as áreas visíveis na tela, e o usuário pode escolher se ficarão visíveis ou ocultas. Ao fechar o Remoto Serial, quando houver ocorrido alguma alteração nas configurações, é solicitado ao usuário confirmação para salvá-las. Estas definições também serão salvas no arquivo de configuração. Ao iniciar o Remoto Serial, as últimas configurações salvas serão recuperadas.

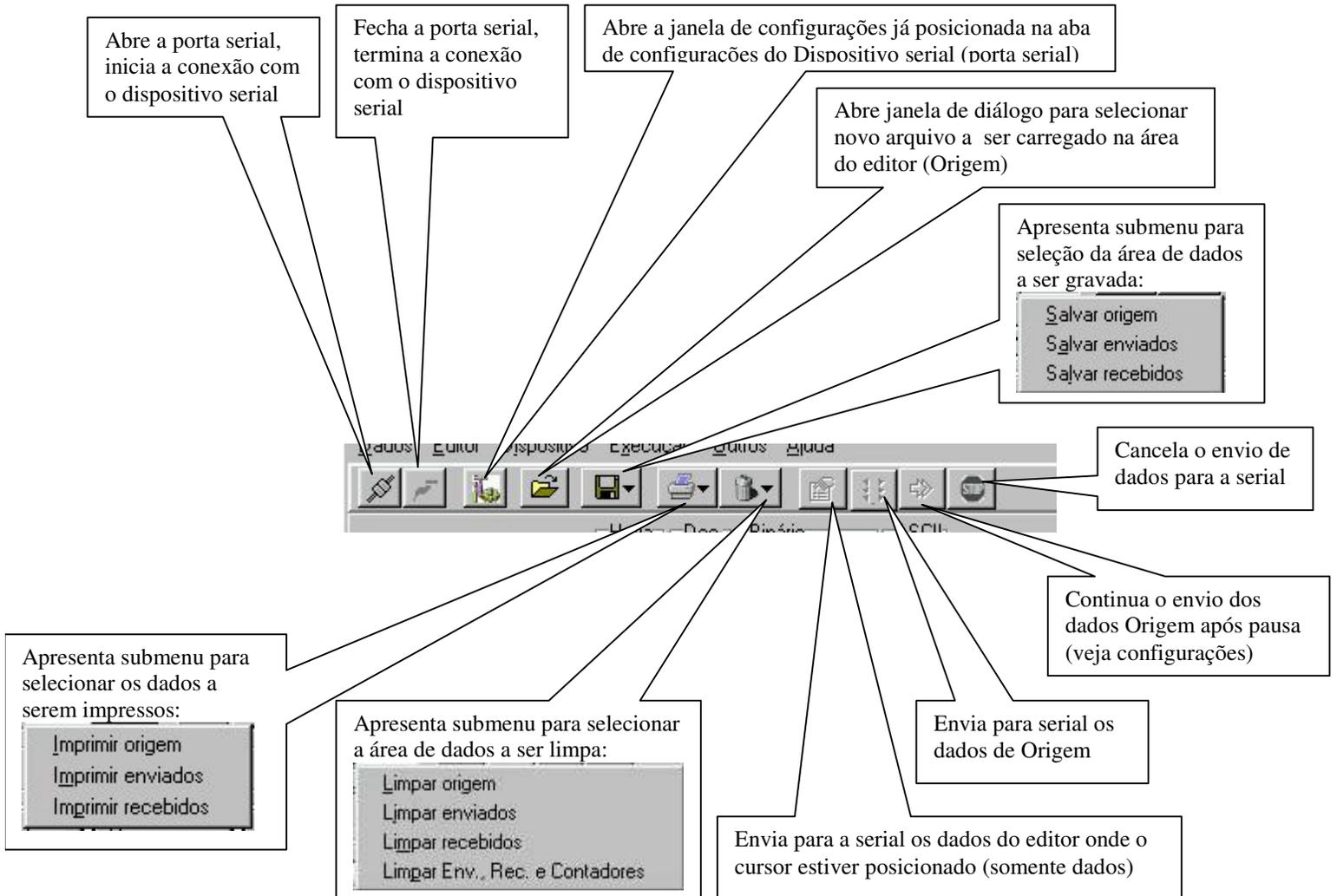


## Menu Ajuda:



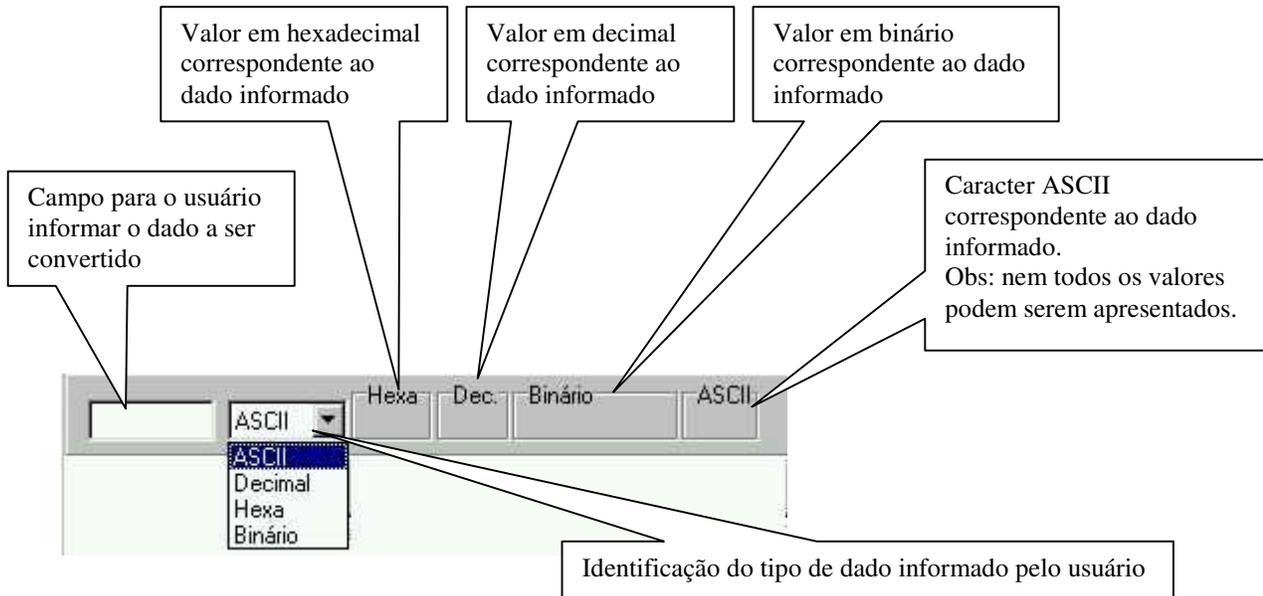
## Barra de Ferramentas:

A barra de ferramentas apresenta alguns botões auxiliares que servem de atalho para as tarefas mais usuais do Remoto Serial, e ainda de alternativa ao menu principal. Como no menu principal, alguns botões só estarão ativos após inicializada a porta serial e outros somente durante o envio de dados.



## Conversor de códigos:

O Conversor de códigos auxilia o usuário a visualizar/converter valores entre bases numéricas e os valores de caracteres ASCII.



### Observações:

- O campo de dados informados pelo usuário aceita somente caracteres condizentes com o tipo/base selecionado no campo Identificação. Por exemplo: ao definir o tipo como binário, o campo de dados somente aceitará caracteres "0" e "1".
- Ao alterar a Identificação de tipo, o dado presente no campo de dados informado pelo usuário será automaticamente convertido para o novo tipo selecionado.
- Alguns valores não possuem caractere ASCII correspondente, nestes casos, será apresentado no campo ASCII um pequeno quadrado negro.

## Editor de texto dos dados de ORIGEM:

O editor é o local onde o usuário escreve os dados a serem enviados pela serial, estes dados também podem ser carregados a partir de arquivos previamente salvos no formato de texto puro.

O usuário pode personalizar a aparência dos diversos tipos de dados (números, strings etc.) através da janela de configuração na aba Editor. As últimas definições do editor salvas (cor, estilo etc.) serão recuperadas na próxima inicialização do Remoto Serial PC.

Área livre do editor para o usuário digitar/alterar os dados de Origem

```
1 // Definição de constantes dos registros do PIC16F877A
2
3 Const
4 // banco 0
5 INDF      0 0
6 TMR0     0 1 // registro do módulo Timer0
7 PCL      0 h'02' // Byte menos significativo do Program Counter (ponteiro do pro
8 // estado 0 h'02' // TRIS - TRIS
```

Área de numeração das linhas do arquivo Origem (automático). Pode ser configurado para não ficar visível.

Clicando-se com o botão direito do mouse sobre a área do editor abre-se um menu popup com opções de ações a serem executadas com os dados:

- Copiar
- Colar
- Desfazer
- Recortar
- Imprimir

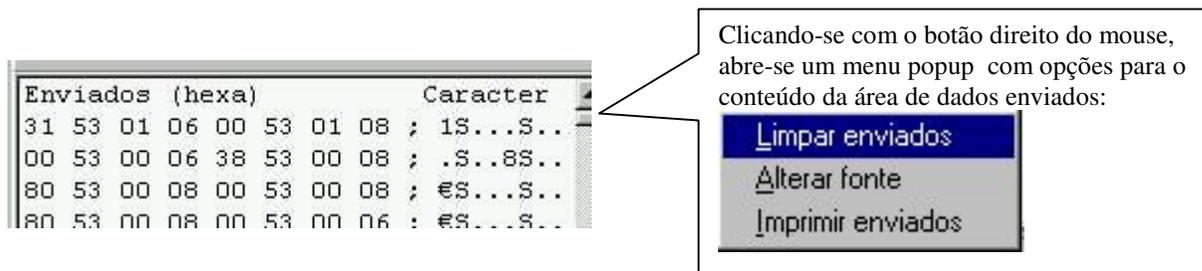
## Área dos bytes efetivamente enviados:

Esta área apresenta os bytes que realmente foram enviados para a porta serial na mesma ordem do envio.

Os bytes são apresentados no formato hexadecimal, e quando possível, seu caracter ASCII correspondente. Quando não existir um caracter ASCII “imprimível”, será usado um ponto .

Nem todos os dados de Origem são enviados para a serial, como : comentários, macros, definições etc.

Está disponível também uma macro que possibilita enviar texto para a área de bytes enviados (veja Macros do Remoto Serial PC), porém o texto não é enviado para a serial, apenas para esta área.



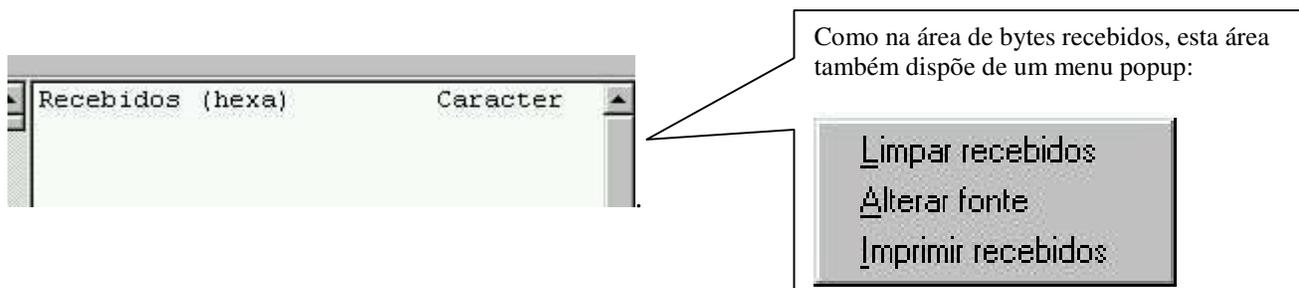
## Área dos bytes recebidos pela serial:

Esta área apresenta os bytes recebidos pela porta serial.

Os bytes são apresentados no formato hexadecimal, e quando possível, seu caracter ASCII correspondente, quando não for possível, será apresentado um ponto na posição do caracter.

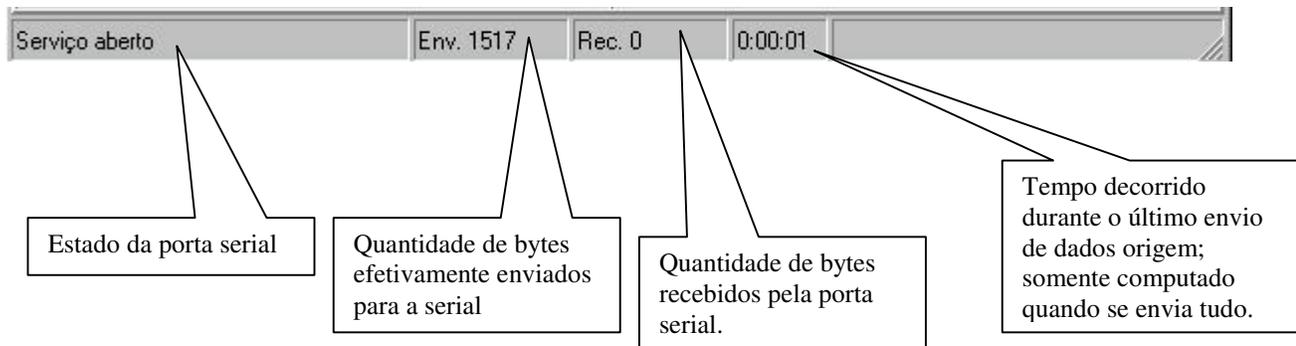
Está disponível também uma macro que possibilita enviar texto para a área de bytes recebidos (veja Macros do Remoto Serial PC).

Dica: Quando se envia um texto para a área de bytes recebidos para identificar um retorno da serial, é aconselhável incluir uma pausa após o comando que irá gerar o retorno, de modo que haja tempo do dado chegar pela serial antes de escrever outro texto.



## Barra de Status:

A barra de status apresenta ao usuário algumas informações relativas a comunicação com a porta serial e o estado da mesma.



Os contadores de bytes enviados e bytes recebidos podem ser zerados clicando-se no menu Outros / Zerar contadores enviados/recebidos.

Existe nas configurações opção de zerar ambos os contadores a cada vez que se inicia o envio de dados.

## Configurações:

A janela de configurações apresenta 3 abas com diversas opções para o usuário configurar o dispositivo serial, o editor de texto dos dados Origem e os parâmetros de execução durante o envio de dados para a serial.

### Configurações do dispositivo serial:

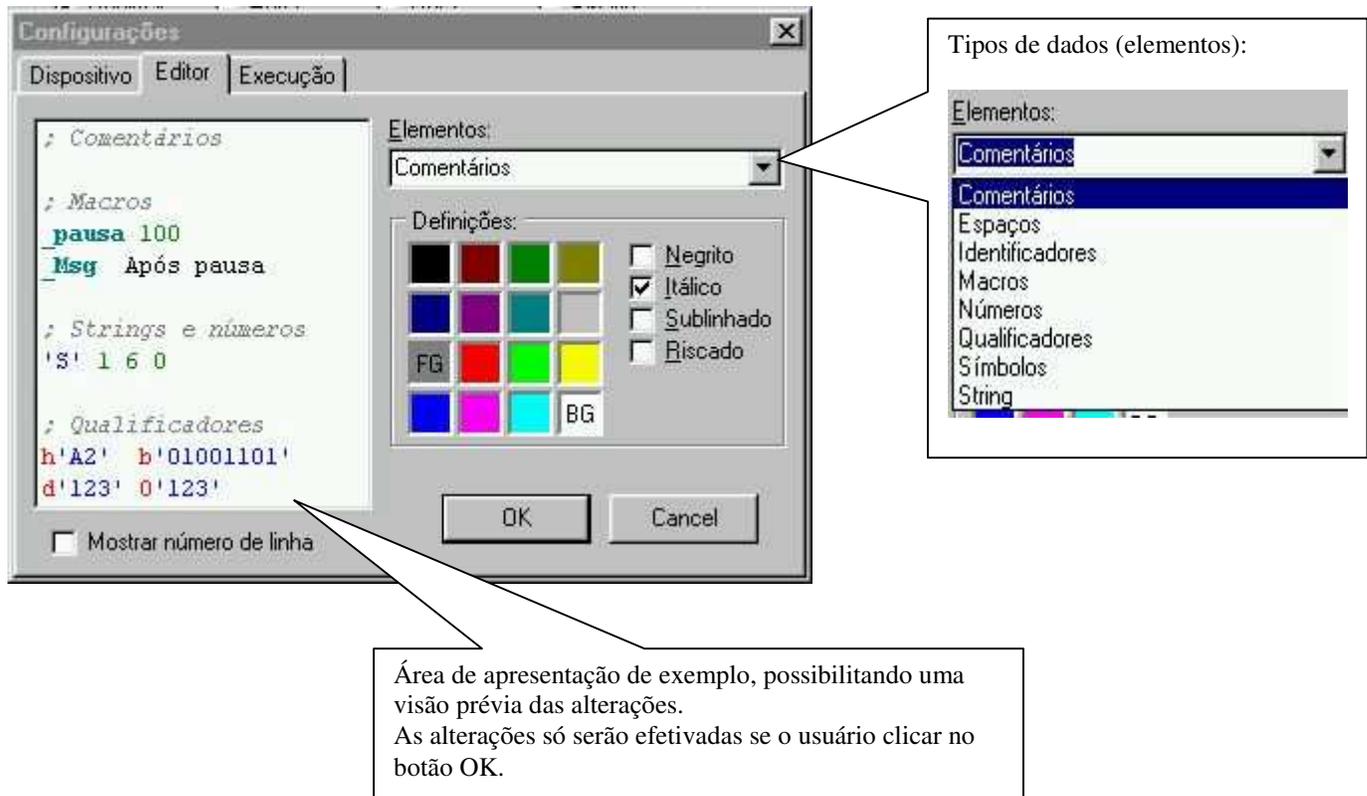
A figura abaixo apresenta as configurações default para a comunicação com o Remoto Serial PIC. Excluindo-se a seleção da porta serial desejada, os demais parâmetros quando alterados demandarão alterações no programa fonte do Remoto Serial PIC ou algum envio de dados específicos para o PIC, antes dos mesmos serem alterados.



Obs.: O Remoto Serial PC ao ser iniciado procura automaticamente as portas seriais disponíveis na máquina do usuário, porém no Windows NT (®) e provavelmente no 2000 estes dados não estão disponíveis no mesmo local. Assim, o Remoto Serial PC fica impossibilitado de identificá-las. Neste caso, antes de aparecer a janela principal do programa será enviada uma mensagem ao usuário - Não foi possível identificar os dispositivos seriais . Verifique as configurações assumidas.

## Configurações do editor

Nas configurações do editor o usuário poderá personalizar a aparência do texto da área do editor do arquivo Origem, como cor de caracteres e atributos (sublinhado, negrito etc.) para cada tipo de dados, e a apresentação ou não da numeração de linhas.



As configurações do editor definidas pelo usuário serão salvas ao término do programa (se o usuário confirmar que quer salvar!), e serão recuperadas na próxima vez que o programa for iniciado.

## Configurações da execução:

As configurações da execução permitem ao usuário diversos “truques”.

The image shows a screenshot of the 'Configurações' (Settings) dialog box, specifically the 'Execução' (Execution) tab. The dialog has three tabs: 'Dispositivo', 'Editor', and 'Execução'. The 'Execução' tab is active and contains several options:

- Carregar enviados
- Limpar bytes enviados, bytes recebidos e contadores no início do envio
- Mensagens:**
  - Apresentar normalmente
  - Ignorar
  - Substituir por pausa de 500 ms
  - Substituir por enviar
- Pausa entre bytes (with a spin box set to 0 ms)
- Pausa entre linhas (with a spin box set to 0 ms)
- OK button

Five callout boxes provide detailed explanations for these options:

- Top-left callout:** Deixando esta opção desmarcada pode aumentar um pouco a velocidade de envio dos dados.
- Top-middle callout:** Marcando-se essa opção, o usuário tem dados exatos do que acontece em cada execução isoladamente.
- Top-right callout:** Permite ao usuário definir uma pausa em milésimos de segundos entre o envio de cada byte para a serial, facilitando o acompanhamento passo a passo.
- Middle-right callout:** Permite ao usuário definir uma pausa em milésimos de segundos entre o envio de cada linha de dados. Marcando-se essa opção com a pausa em 0 (zeros) milésimos, a execução será interrompida entre cada linha, possibilitando o acompanhamento dos efeitos a cada linha. Para passar para a próxima linha, o usuário deverá clicar no botão continue na barra de ferramentas ou na opção correspondente do menu Execução.
- Bottom callout:** Permite ao usuário definir o que o programa fará ao encontrar uma macro de mensagem no arquivo Origem durante o envio de dados.

As macros de mensagem são muito úteis durante a depuração de um arquivo origem, identificando exatamente o local do arquivo Origem que está sendo executado (dados a serem enviados). Porém desejando-se uma execução completa, a confirmação de cada mensagem pode ser um incômodo, bem como ter que transformar todas as mensagens em comentários. Para evitar esses transtornos basta alterar a configuração do tratamento dado pelo programa a estas macros .

## Comandos do Remoto Serial PC:

Como o objetivo do Remoto Serial é enviar bytes pela serial, os comandos basicamente são valores de forma textual que são “lidos” e enviados para a porta.

### - Sintaxe dos dados para serem enviados:

É necessário pelo menos um espaço entre cada dado a ser enviado.

Os dados contidos na área Origem são interpretados da seguinte forma:

**;** (**ponto e vírgula**)- comentário, tudo na mesma linha que vier depois do ponto e vírgula será ignorado (veja limitações).

**h'..' ou H'..' -** valor hexadecimal, os caracteres A até F que compõem o valor hexadecimal podem estar em maiúsculas, minúsculas, ou misturado. Ex. **h'Ff'** = 255 (decimal)

**b'.....' ou B'.....' -** valor binário. Ex. **b'10101010'** = 170 (decimal)

**d'...' ou D'...' -** valor decimal (default). Ex. **d'23'**

**o'...' ou O'...' -** valor octal. Ex. **o'12'** = 10 (decimal)

**'.....' –** caracteres. Ex. **'Teste'** = 84 101 115 116 101 (decimal)

54 65 73 74 65 (hexadecimal)

A base para valores numéricos default é a decimal, assim, estes não precisam ser colocados entre aspas nem do prefixo 'd' (ou 'D'), ou seja, com a base decimal, 123 é a mesma coisa que D'123'. O mesmo acontece para valores nas outras bases quando estas estiverem marcadas na área de Origem como Base default, por exemplo: definindo a base default como hexa, A5 é a mesma coisa que h'A5', deste modo, valores em bases diferentes da default necessitam estar entre aspas e com o devido prefixo.

## Macros do Remoto Serial PC:

Visando facilitar a construção dos arquivos de dados a serem enviados e evitando-se digitar várias vezes os mesmos dados, etc. , o Remoto Serial dispõe das seguintes macros:

### Nota importante:

Todos os parâmetros de valores obedecem a base default, assim, é aconselhável utilizar-se sempre o qualificador de base para evitar resultados indesejados ou mesmo mensagens de erro ao alterar-se a base. Ex.: 11(hexadecimal) = 17 (decimal), 11(octadecimal) = 9 (decimal) , 11(binário) = 3 (decimal) ; 134 é um valor válido para bases hexa, decimal e octal (válido mas com “pesos” totalmente diferentes) porém com a base em binário ocasionará erro.

### ***\_repetir qtd***

...

### ***-fimrepetir***

Repete a seqüência "qtd" vezes, sendo no mínimo 1 vez, mesmo com qtd=0; podem ser utilizadas quantas vezes quiser no arquivo, porém só podem ser aninhadas no máximo 10 ao mesmo tempo.

Ex:

*\_repetir 5*

*....(trecho1) ;trecho do arquivo que será repetido 5 vezes*

*\_repetir 2*

*....(trecho2) ;trecho do arquivo que será repetido 10 vezes (5 x 2)*

*..... ; intercalados com o trecho1*

*\_fimrepetir*

*\_fimrepetir*

### ***\_MSG*** Mensagem a ser mostrada

Pára a transmissão dos dados e abre uma janela com a mensagem informada, basta teclar enter ou clicar no botão OK da mensagem para continuar.

Ex:

*\_MSG Configurar LCD para 4 bits*

### ***\_SMSG*** Mensagem a ser mostrada

O mesmo que *\_msg*, porém acompanhado de um sinal sonoro (beep).

### ***\_PAUSA qtd***

Efetua uma pausa na transmissão de "qtd" milésimos de segundos.

EX:

*\_pausa 500 ; efetua uma pausa de ½ segundo.*

**\_V00 vl\_inicial vl\_incremento/decremento**  
**\_V01 vl\_inicial vl\_incremento/decremento**  
**\_V02 vl\_inicial vl\_incremento/decremento**  
**\_V03 vl\_inicial vl\_incremento/decremento**  
(no momento está restrito a estas 4 variáveis)

Declara a variável para ser usada dinamicamente dentro do arquivo. As variáveis podem ser redefinidas em qualquer instante. Podem ser usadas como dado a ser transmitido , para compor mensagens , para parâmetro de pausa e até parâmetro para definição de outra variável. Tanto o valor inicial quanto o valor de incremento/decremento quando não qualificados, obedecem a base numérica atual.

Ex:

```
_V00 32 16 ; define a variável 00 com o valor inicial de 32 e  
;;;;;;;;;; 16 para incremento e/ou decremento
```

### **\_I00 \_I01 \_I02 \_I03**

Incrementa o valor da respectiva variável com o valor indicado na última declaração da mesma.

### **\_D00 \_D01 \_D02 \_I03**

Decrementa o valor da respectiva variável com o valor indicado na última declaração da mesma.

### **\_U00 \_U01 \_U02 \_U03**

Utilização do valor atualmente contido na respectiva variável

obs: a macro será substituída pela notação numérica do conteúdo da variável na base atual (até a versão 1.2 os valores eram apresentados na notação hexadecimal com o respectivo qualificador).

Ex:

```
_V00 32 16 ;define V00, vlr inicial 32(h'20') inc/dec 16 (h'10')  
'S' 0 6 _U00 ; será transmitido h'53' h'00' h'06' h'20'  
_MSG Valor atual de 00 : _U00  
;;; aparecerá a janela com --> Valor atual de 00 : 32  
_I00  
_MSG Valor atual de 00 : _U00  
;;; aparecerá a janela com --> Valor atual de 00 : 48  
_V00 160 10 ;V00 foi redeclarada  
_D00 ; V00 passa a conter 150  
_D00 ; V00 passa a conter 140  
_I00 ; V00 passa para 150  
_D00 ; V00 passa para 140  
_MSG Valor atual de 00 : _U00  
;;; aparecerá a janela com --> Valor atual de 00 : 140
```

## **\_AGUARDARPOR *qtd***

...

## **\_FIMAGUARDAR**

Fica repetindo o trecho entre `_AguardarPor` e `_FimAguardar` indefinidamente até que seja recebido pela serial pelo menos “qtd” bytes.

Ex:

```
_aguardarpor 10
'E' 2 'OK' ; no remoto pic, palavra 'OK' é ecoada de volta para o PC
_pausa 100
_fimaguardar
```

O trecho será repetido 5 vezes, visto que cada vez o PIC devolve 2 bytes (OK) .

É recomendado colocar-se uma pausa ( `_pausa` ) ou uma mensagem ( `_MSG` ) antes de `_fimaguardar`, de modo a dar tempo do PIC responder ao/aos comandos antes de repetir o envio para o PIC.

## **\_Beep**

Emite um sinal sonoro no PC porém não pára o processamento. Obs: é necessário um certo tempo para o PC emitir o sinal e estar preparado para emitir novo sinal, assim, se for encontrado uma ou mais macros `_beep` antes do sinal sonoro anterior terminar, o PC não emitirá os outros sinais.

Ex:

```
_repetir 5
_beep
_fimrepetir
;----> só será ouvido um sinal ( a não ser que esteja definido pausa entre linhas no respectivo campo
;; da tela de configuração!)
```

```
_repetir 3
_beep
_pausa 500
_fimrepetir
;----> serão ouvidos 3 sinais sonoros (3 beeps).
```

## **\_EcoEnv Texto a ser exibido**

Insere o texto desejado na área de bytes enviados. Esta macro é útil para documentar os dados na área de bytes enviados, bem como auxiliar na depuração.

Ex:

```
_ecoenv Definir TRISA como output
“S” 1 h'05' 0
```

Será apresentado na área de bytes enviados:

```
; Definir TRISA como output
53 01 05 00 ; S...
```

## **\_EcoRec** Texto a ser exibido

A mesma função que `_EcoEnv`, porém o texto é exibido na área de bytes recebidos. Esta macro é muito útil para identificar dados que foram recebidos após um comando de leitura.

Ex:

```
_Ecoenv Valor atual de PORTA
"L" 0 D'05'
_pausa 200          ; dica: como o processamento no PC pode ser mais rápido que a resposta do
                   ; PIC, uma pausa evita que o texto do _ecoenv seguinte seja exibido na área de
                   ; bytes recebidos antes do retorno do valor solicitado.
_EcoEnv Valor atual de PORTB
"L" 0 D'06'
```

## **\_Define** *nome\_para\_a\_macro*

...

...

## **\_FimDefine**

A macro `_define` permite ao usuário criar suas próprias macros contendo vários comandos/dados. É possível definir até 500 macros por cada execução. Se for definida uma macro com o mesmo nome de uma já previamente definida, a última definição terá prioridade.

Ex:

```
_Define pulso_portb0
"S" 0 h'06' b'00000001'
"S" 0 h'06' b'00000000'
_fimdefine

;;;;; executar a macro
pulso_portb0
```

## **\_Const**

*nome1 valor de nome1*

*nome2 valor de nome2*

...

## **\_FimConst**

A macro `_const` permite ao usuário definir uma ou mais constantes, possibilitando assim uma forma mais amigável para escrever os arquivos Origem. É possível criar até 500 constantes por execução. Enquanto na definição de macros tem que ser utilizado um `_Define` para cada declaração e esta pode conter mais de uma linha, na declaração de constantes pode-se utilizar apenas uma macro `_const` para definir várias constantes, porém cada constante não pode ocupar mais de uma linha.

Ex:

```
_const
ler "L"          ; constante ler contém 4C (hexadecimal)
set "S"          ; constante set contém 53 (hexadecimal)
TRISA 1 h'05'    ; constante trisa contém 01 05 (hexadecimal)
PORTA 0 h'05'    ; constante porta contém 00 05 (hexadecimal)
TRISB 1 h'06'    ; constante trisb contém 01 06 (hexadecimal)
PORTB 0 h'06'    ; constante portb contém 00 06 (hexadecimal)
_fimconst

Set TrisA d'00'  ; define TRISA para PORTA tudo output
Set TRISB h'FF' ; define TRISB para PORTB tudo input
Set PortA b'10101010' ; envia AA (hexadecimal) para PORTA
ler portB       ; lê o valor presente em PORTB (o valor será apresentado na área
                 ; de bytes recebidos)
```

## **\_Incluir arquivo\_a\_ser\_incluido**

Permite ao usuário criar rotinas em módulos que serão incluídos dinamicamente durante a execução. É possível incluir até 100 arquivos por execução. O arquivo incluído pode conter dados, macros, definições de constantes ,etc. até mesmo outros arquivos incluídos (respeitando-se o limite total de 100).

Ex.:

```
_incluir "registros.ori"
_incluir "d:\arqsori\LCDs\16x4.ori"
_const
ler "L"
set "S"
_fimconst

iniciar_lcd      ; macro contida em 16x4.ori
Set TRISA H'00' ; constante TrisA definida em registros.ori
"S" PORTA h'ff" ; ativar todos os pinos de PORTA (definida em registros.ori)
```

## **DICAS:**

**- É possível alterar a taxa de transmissão do PIC remoto (default 19200) , para tal, envie somente a linha mostrada abaixo e depois mude a taxa na janela de configurar dispositivo do Remoto (PC) para a nova taxa definida:**

**'S' 1 25 xx ;; o 25 está em decimal**

**Onde xx é o valor que define a taxa de transmissão da USART do PIC (veja datasheet dos PICs).**

**- É aconselhável utilizar-se sempre o qualificador de base para valores numéricos, deste modo evitam-se problemas ao trocar-se a base definida como default.**

**- Utilizar valores na base binária facilita identificar os bits ativos.**

**- Após a execução de um arquivo Origem, os dados presentes na área de bytes enviados podem ser copiados para criar um novo arquivo Origem na base hexadecimal, deste modo todo o trabalho de desenvolvimento fica mais “oculto”. Utilize vários “1” seguidos para simular pausas.**

## Comandos do Remoto Serial PIC:

O Remoto Serial PIC fica constantemente verificando a serial em busca de algum comando e seus parâmetros quando necessário.

**“0” (caracter zero, 30h, 48 decimal)** - Reset do PIC - apaga LED de erro, ativa confirmação para todos os comandos recebidos (mensagem OK) e define a maioria dos pinos dos Port's para output.

**“1” (caracter 1, 31h, 49 decimal)** - Desativa confirmação dos comandos, ou seja, não será devolvida a mensagem OK após os comandos.

**“2” (caracter 2, 32h, 50 decimal)** - Ativa confirmação dos comandos, ou seja, será devolvida a mensagem OK após os comandos.

**“6” (caracter 6, 36h, 54 decimal)** - Versão - Devolve para o PC o nome e versão do programa.

**“S” (caracter S, 53h, 83 decimal)** - Set - Carrega o valor recebido no registro especificado. Recebe 3 parâmetros: Banco, Endereço do registro(offset) e Valor.

Ex.: 'S' 1 6 0 ; Setar TRISB output

**“L” (caracter L, 4Ch, 76 decimal)** - Load - Devolve para o PC o valor contido no registro solicitado. Recebe 2 parâmetros: Banco e Endereço do registro (offset).

Ex.: 'L' 0 8 ; Ler o valor presente no PORTD

**“E” (caracter E, 45h, 69 decimal)** - Eco - Ecoa os n bytes recebidos de volta para o PC (limitado ao máximo de 48 bytes). Recebe 2 parâmetros: Quantidade de bytes, Bytes para ecoar .

Ex.: 'E' 5 'TESTE' ; Ecoa "TESTE" de volta para o PC.

Qualquer comando diferente dos acima é considerado inválido, o PIC devolve ao PC a mensagem Erro ! e acende o LED de aviso (se estiver implementado e o Port não tiver sido alterado por algum comando).

## **Possíveis implementações futuras:**

A seguir algumas idéias para implementações tanto no Remoto Serial PC quanto no Remoto Serial PIC; essas idéias podem ser incluídas em novas versões, ou mesmo nunca serem implementadas.

### **No Remoto Serial PC:**

- Macro para alterar dinamicamente a base default durante o processamento.
- Macro para alterar a taxa de transmissão (baud rate) do Remoto Serial PC durante o processamento.

### **No Remoto Serial PIC:**

- Comando para enviar vários bytes (em lote) para um mesmo endereço no PIC.
- Comando para definir bit's independentes em qualquer endereço no PIC.
- Comando para efetuar operações lógicas (AND, OR etc.) com qualquer endereço no PIC.
- Comando para devolver o conteúdo de n endereços consecutivos do PIC.
- Comando para devolver o conteúdo de um endereço n vezes seguidas.



## Exemplos:

```
;Arquivo: Registros.ori
;Autor : Renie S. Marquet
;Data : 18.09.2003
;Descrição:
; Definição de constantes dos registros do PIC16F877A
```

```
_Const
; banco 0
; banco endereço (offset)
INDF 0 0
TMR0 0 1 ; registro do módulo Timer0
PCL 0 h'02' ; Byte menos significativo do Program Counter (ponteiro do programa)
STATUS 0 h'03' ; IRP ; RP1 ; RP0 ; -TO ; -PD ; Z ; DC ; C
FSR 0 h'04' ; Ponteiro indireto da memória de dados
PORTA 0 h'05' ; - ; - ; Latch do PORTA quando escrito, pinos do PORTA quando lido
PORTB 0 h'06' ; Latch do PORTB quando escrito, pinos do PORTB quando lido
PORTC 0 h'07' ; Latch do PORTC quando escrito, pinos do PORTC quando lido
PORTD 0 h'08' ; Latch do PORTD quando escrito, pinos do PORTD quando lido
PORTE 0 h'09' ; - ; - ; - ; - ; - ; Latch do PORTE quando escrito, pinos do PORTE
; quando lido
PCLATH 0 h'0A' ; - ; - ; - ; buffer para 5 bits superiores do Program Counter
INTCON 0 h'0B' ; GIE ; PEIE ; TMR0IE ; INTE ; RBIE ; TMR0IF ; INTF ; RBIF
PIR1 0 h'0C' ; PSPIF ; ADIF ; RCIF ; TXIF ; SSPIF ; CCP1IF ; TMR2IF ; TMR1IF
PIR2 0 h'0D' ; - ; CMIF ; - ; EEIF ; BCLIF ; - ; - ; CCP2IF
TMR1L 0 h'0E' ; guarda byte menos significativo do registro de 16 bits TMR1
TMR1H 0 h'0F' ; guarda byte mais significativo do registro de 16 bits TMR1
T1CON 0 h'10' ; - ; - ; T1CKPS1 ; T1CKPS0 ; T1OSCEN ; -T1SYNC ; TMR1CS ; TMR1ON
TMR2 0 h'11' ; registro do módulo do Timer2
T2CON 0 h'12' ; - ; TOUTPS3 ; TOUTPS2 ; TOUTPS1 ; TOUTPS0 ; TMR2ON ; T2CKPS1 ; T2CKPS0
SSPBUF 0 h'13' ; buffer para porta serial síncrona
SSPCON 0 h'14' ; WCOL ; SSPOV ; SSPEN ; CKP ; SSPM3 ; SSPM2 ; SSPM1 ; SSPM0
CCPR1L 0 h'15' ; registro1 (byte baixo) de capture/compare/PWM
CCPR1H 0 h'16' ; registro1 (byte alto) de capture/compare/PWM
CCP1CON 0 h'17' ; - ; - ; CCP1X ; CCP1Y ; CCP1M3 ; CCP1M2 ; CCP1M1 ; CCP1M0
RCSTA 0 h'18' ; SPEN ; RX9 ; SREN ; CREN ; ADDEN ; FERR ; OERR ; RX9D
TXREG 0 h'19' ; registro de dados para transmissão da USART
RCREG 0 h'1A' ; registro de dados para recebimento da USART
CCPR2L 0 h'1B' ; registro2 (byte baixo) de capture/compare/PWM
CCPR2H 0 h'1C' ; registro2 (byte alto) de capture/compare/PWM
CCP2CON 0 h'1D' ; - ; - ; CCP2X ; CCP2Y ; CCP2M3 ; CCP2M2 ; CCP2M1 ; CCP2M0
ADRESH 0 h'1E' ; byte alto do resultado do conversor A/D
ADCON0 0 h'1F' ; ADCS1 ; ADCS0 ; CHS2 ; CHS1 ; CHS0 ; GO/-DONE ; - ; ADON
;; banco 1
; banco endereço (offset)
OPTION_REG 1 1 ; -RBPU ; INTEDG ; T0CS ; T0SE ; PSA ; PS2 ; PS1 ; PS0
TRISA 1 h'05' ; - ; - ; Registro de direção dos dados do PORTA
TRISB 1 h'06' ; Registro de direção dos dados do PORTB
TRISC 1 h'07' ; Registro de direção dos dados do PORTC
TRISD 1 h'08' ; Registro de direção dos dados do PORTD
TRISE 1 h'09' ; IBF ; OBF ; IBOV ; PSPMODE ; - ; Registro de direção dos dados do
; PORTE (3 bits)
PIE1 1 h'0C' ; PSPIE ; ADIE ; RCIE ; TXIE ; SSPIE ; CCP1IE ; TMR2IE ; TMR1IE
PIE2 1 h'0D' ; - ; CMIE ; - ; EEIE ; BCLIE ; - ; - ; CCP2IE
PCON 1 h'0E' ; - ; - ; - ; - ; - ; - ; -POR ; -BOR
OSCCAL 1 h'0F'
SSPCON2 1 h'11' ; GCEN ; ACKSTAT ; ACKDT ; RCEN ; PEN ; RSEN ; SEN
PR2 1 h'12' ; Registo do módulo TIMER2
SSPADD 1 h'13' ; Registro da porta serial (modo I2C) síncrona
SSPATAT 1 h'14' ; SMP ; CKE ; D/-A ; P ; S ; R/-W ; UA ; BF
TXSTA 1 h'18' ; CSRC ; TX9 ; TXEN ; SYNC ; - ; BRGH ; TRMT ; TX9D
SPBRG 1 h'19' ; Registro do gerador de Baud Rate (taxa de transmissão)
CMCON 1 h'1C' ; C2OUT ; C1OUT ; C2INV ; C1INV ; CIS ; CM2 ; CM1 ; CM0
```

```

CVRCON      1 h'1D' ; CVREN ; CVROE ; CVRR ; - ; CVR3 ; CVR2 ; CVR1 ; CVR0
ADRESL      1 h'1E' ; byte baixo do resultado do conversor A/D
ADCON1      1 h'1F' ; ADFM ; ADC2 ; - ; - ; PCFG3 ; PCFG2 ; PCFG1 ; PCFG0
;; banco 2
; banco endereço (offset)
PORTF h'02' h'07' ; não presente no F877
PORTG h'02' h'08' ; não presente no F877
EEDATA h'02' h'0C' ; byte baixo do registro de dados da EEPROM
EEADR h'02' h'0D' ; byte baixo do registro de endereço da EEPROM
EEDATH h'02' h'0E' ; - ; - ; byte alto do registro de dados da EEPROM
EEADRH h'02' h'0F' ; - ; - ; - ; - ; byte alto do reg. endereço da EEPROM
;; banco 3
; banco endereço (offset)
TRISF h'03' h'07' ; não presente no F877
TRISG h'03' h'08' ; não presente no F877
EECON1 h'03' h'0C' ; EEPGD ; - ; - ; WRERR ; WREN ; WR ; RD
EECON2 h'03' h'0D' ; Registro 2 de controle da EEPROM (não é um registro físico)

```

```
_fimconst
```

```
-----x-x-x-x-xxxxxxxxxxxxxxxxxxxxxxxx-----x-x-x-x-----
```

```

;Arquivo : Sequencial.ori
;Autor   : Renie S. Marquet
;Data    : 14.10.2003
;Descrição : Corre um único led aceso no PORTB
; para executar este teste, ligue um resistor de 220 ohms e um Led
; entre cada pino do PORTB (pinos 33 a 40) e o terra
; pino portb ---/\/\\/\ --- |>|---terra
;                220 R      Led

```

```
_incluir 'registros.ori'
```

```
_const
```

```
Ler "L"
```

```
Set "S"
```

```
_fimconst
```

```
_define Wait
```

```
_pausa 500 ; 500 milésimos de seg. (1/2 seg.)
```

```
_fimdefine
```

```
set TrisB 0 ; define PORTB output
```

```
_repetir 10 ; a seqüência abaixo será repetida 10 vezes
```

```
set PORTB 0 ; colocar todos os pinos do PORTB em nível 0
```

```
Wait ; macro para pausa de ½ seg.
```

```
Set portb b'00000001' ;
```

```
wait
```

```
set portB b'00000010' ; Como o Remoto Serial não é case sensitive,
```

```
wait ; qualquer combinação de maiúsculas e minúsculas
```

```
set PORTb b'00000100' ; são válidas como PORTB = portb = PORTb = PortB...
```

```
wait
```

```
set portb b'00001000' ;
```

```
wait
```

```
set portb b'00010000' ; usar a notação binária facilita identificar os
```

```
wait ; bit's do port que estarão ativos
```

```
set portb b'00100000' ;
```

```
wait
```

```
set portb b'01000000' ;
```

```
wait
```

```
set portb b'10000000' ;
```

```
_fimrepetir ; final do arquivo
```

```

;Arquivo : Sequencial_2.ori
;Autor   : Renie S. Marquet
;Data    : 14.10.2003
;Descrição : Efeito barra de Leds no PORTB com uso de variáveis calculadas
; para executar este teste, ligue um resistor de 220 ohms e um Led
; entre cada pino do PORTB (pinos 33 a 40) e o terra
; pino portb ---/\/\\/ ---- |>|---terra
;                220 R      Led

_incluir 'registros.ori'

_const
Ler "L"
Set "S"
_fimconst

_define Wait
_pausa d'500' ; 500 milésimos de seg. (1/2 seg.)
_fimdefine

set TrisB 0 ; define PORTB output

_repetir d'10' ; a seqüência abaixo será repetida 10 vezes

_V00 1 1 ; define variável v00 inicial 1, incremento 1
_repetir d'8' ; para correr os 8 bits

Set PORTB _U00 ;
_ecorec _u00
_V01 _U00 1 ; guarda valor atual de v00 em v01
_I01 ; incrementa v01 de 1
_V02 _U00 _U01 ; copia v00 em v02, define o incremento = v00 + 1
_I02 ;
_V00 _U02 1 ; redeclara v00 para o novo valor = 2 x V00 + 1
wait
_fimrepetir ; final do repetir 8
_fimrepetir ; final do repetir 10

```