

Programação em BASIC para o PIC

Comunicação Serial

Vitor Amadeu Souza

Introdução

Nesta quarta série relatando o uso do mikroBASIC para programar o PIC, veremos como realizar a comunicação serial entre o PC e o microcontrolador. A idéia será ligar ou desligar um relé instalado na placa didática, de acordo com o caracter recebido via comunicação serial. Esta aplicação seria bem interessante para no caso de alguma automação residencial, onde em determinados momentos alguma lâmpada poderia ligar ou desligar de acordo com o horário ajustado no PC. A comunicação utilizada é do tipo RS232, e os níveis lógicos associados ao nível alto e baixo deste tipo de comunicação e do TTL podem ser observados na tabela 1.

Níveis / Tipos	TTL	RS232
1	5 Vcc	-3 a -18V
0	0 Vcc	3 a 18 V

Tabela 1 – Níveis de comunicação

Note que os níveis utilizados pelo PIC são os níveis TTL enquanto o utilizado pelo PC, é o RS232. Para compatibilizar um nível ao outro, é necessário um conversor de níveis de TTL para RS232 e vice-versa. Um conversor muito popular hoje no mercado é o MAX232 desenvolvido pela Maxim. Este conversor pode ser observado na figura 1.

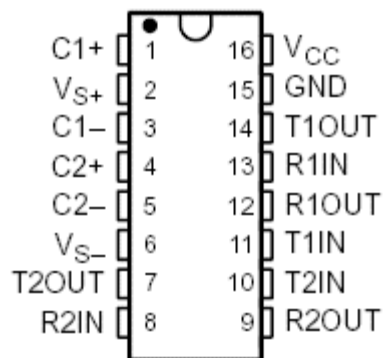


Figura 1 – Conversor TTL – RS232 e vice-versa

O esquema básico de ligação entre o microcontrolador e o PC, de forma com que os mesmos possam se comunicar está apresentado na figura 2.

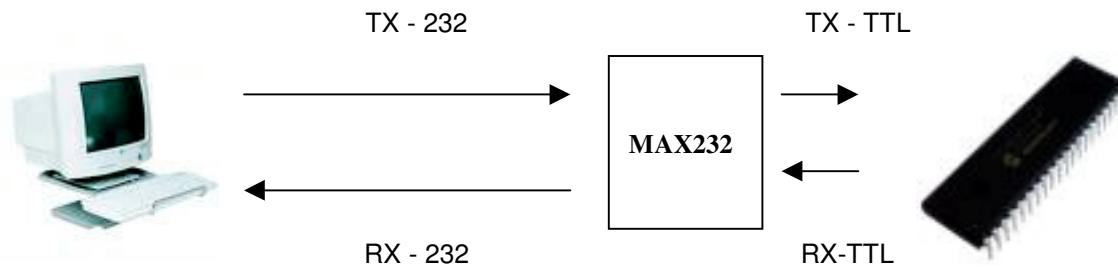


Figura 2 – Esquema de ligação básico para comunicação

Observe que o circuito de conversão, que é o CI MAX232 e o microcontrolador ficam ambos colocados na placa PICLAB18F442.

Para entendermos melhor o funcionamento deste tipo de comunicação, vamos plotar um gráfico que irá apresentar a comunicação no nível TTL e RS232. Verifique a transmissão do byte 01101010 na figura 3.

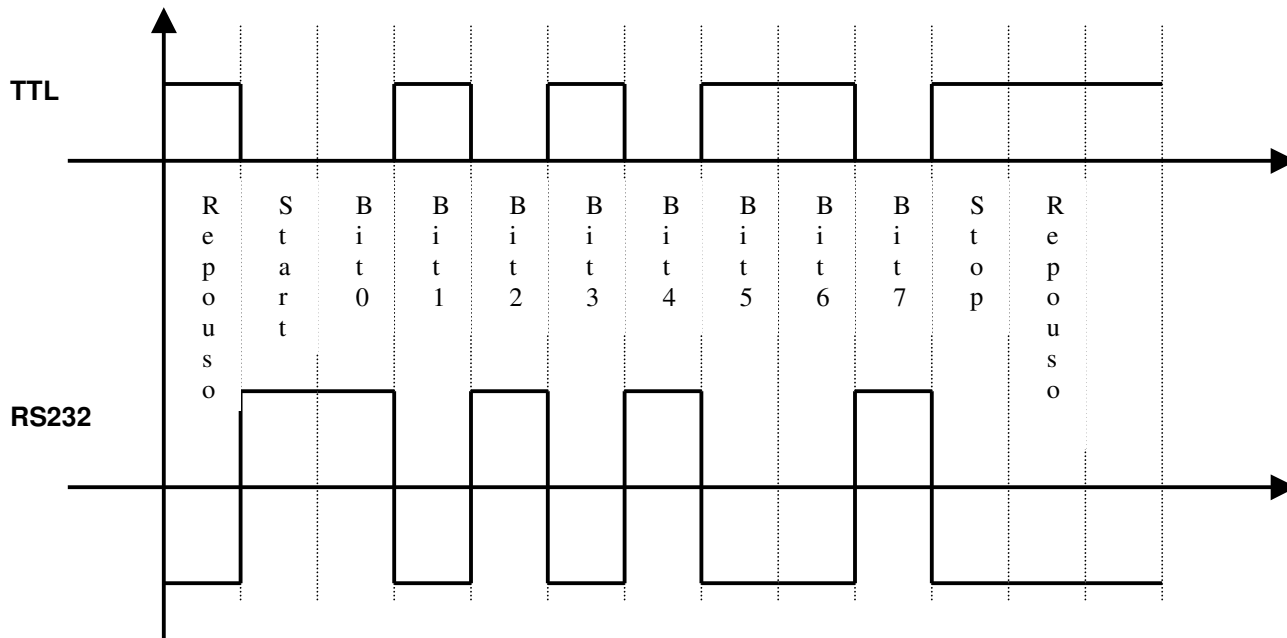


Figura 3 – Plotagem do gráfico na linha TTL e RS232

Observe que no momento em que não há comunicação, a linha TTL permanece em nível alto. Este é o conhecido estado de repouso, ou seja, o momento em que não há comunicação na linha e a mesma está pronta para começar. Toda vez em que um byte será enviado, primeiramente temos o bit de

start (início) e logo em seguida, os oito bits do byte a ser transmitido, começando sempre pelo bit menos significativo. Para finalizar a comunicação, temos o bit de stop (parada) em que neste momento, a linha de dados volta ao seu estado de repouso.

Toda comunicação serial, seja ela USB, RS485 ou RS232 utiliza uma taxa de comunicação (baud rate). Com o baud rate, podemos saber quantos bits podem trafegar pela linha em um intervalo de 1 segundo. Comumente os baud rates são múltiplos de 300 bps (bits por segundo), encontrando desta forma taxas de comunicação como por exemplo 2400 bps, 4800 bps e 9600 bps. Em nosso exemplo, iremos utilizar a taxa de 9600 bps e para acharmos o tempo que 1 bit demora para ser transmitido nesta taxa, basta dividirmos 1 pelo número de bits por segundo. No gráfico plotado na figura 3, considerando o baud rate de 9600 bps, achamos o tempo de aproximadamente de 104 us, pois $1/9600$ é igual a este valor.

Existem basicamente três modos de comunicação utilizados, sendo estes o simplex, half duplex e full duplex. A comunicação simplex é aquela em que se dá em somente 1 sentido. Um exemplo seria o caso da televisão, em que o usuário somente consegue receber dados oriundos da central de transmissão e nunca consegue fazer o processo inverso. Poderíamos representar a comunicação simplex como uma seta de uma só direção, como apresentado na figura 4.



Figura 4 – Representação da comunicação Simplex

Já a comunicação half duplex se dá nos dois sentidos, porém neste caso a linha de comunicação é compartilhada, sendo em instante somente de transmissão e em outro somente de recepção. Existem aparelhos telefônicos, como o da operadora NEXTEL que é um exemplo deste tipo de comunicação. Neste caso, poderíamos representar este tipo de comunicação como apresentado na figura 5, com uma seta única com comunicação bidirecional.



Figura 5 – Representação da comunicação Half Duplex

Já a comunicação full duplex permite a troca de dados simultaneamente, pois temos duas linhas, sendo uma de transmissão e outra de recepção. Um exemplo corriqueiro seria o uso de um celular, em que duas pessoas conseguem falar no mesmo momento e trocar informações. A representação desta comunicação seria como apresentado na figura 6, com uma seta somente de transmissão e outra somente de recepção.

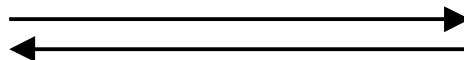


Figura 6 – Representação da comunicação Full Duplex

A comunicação do tipo RS232 é do tipo full duplex, pois temos uma linha somente de transmissão e outra somente de recepção. Desta forma, enquanto estamos transmitindo um byte pela linha de TX, podemos perfeitamente estar recebendo outro pela linha de RX.

Recursos de Hardware

O esquema elétrico deste exemplo pode ser observado na figura 7. Observe que as linhas de comunicação do PIC ficam ligadas ao MAX232 e no pino RB0 está conectado o relé do nosso exemplo. Além disso, o microcontrolador e o MAX232 ficam ambos alimentados através de uma fonte de 5 Vcc.

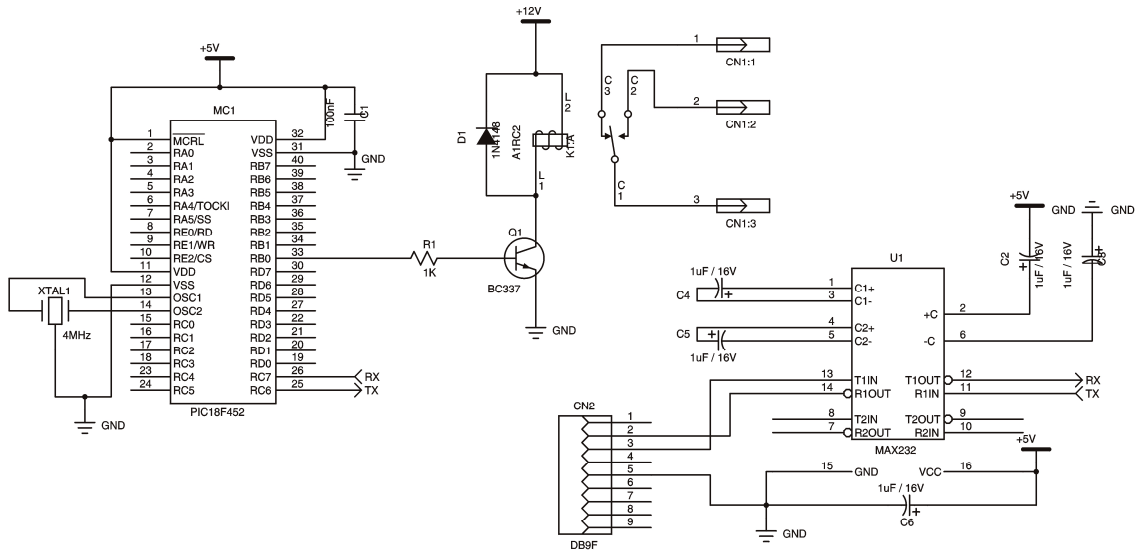


Figura 7 – Esquema Elétrico do Exemplo

Fluxograma

O fluxograma que irá reger este exemplo está apresentado na figura 8. Note que primeiramente é feita a configuração do microcontrolador, como a configuração dos pinos e da USART e logo em seguida o sistema entra em loop infinito verificando a existência de algum byte no canal serial. Ao ser verificado que algum byte foi recebido, é checado se o mesmo é o caracter “A” e caso seja, o relé é acionado. Caso o caracter não seja o “A”, o buffer é novamente testado e caso seja o “B”, o relé é desligado. Quaisquer caracteres diferentes destes dois não terão efeito sobre o sistema.

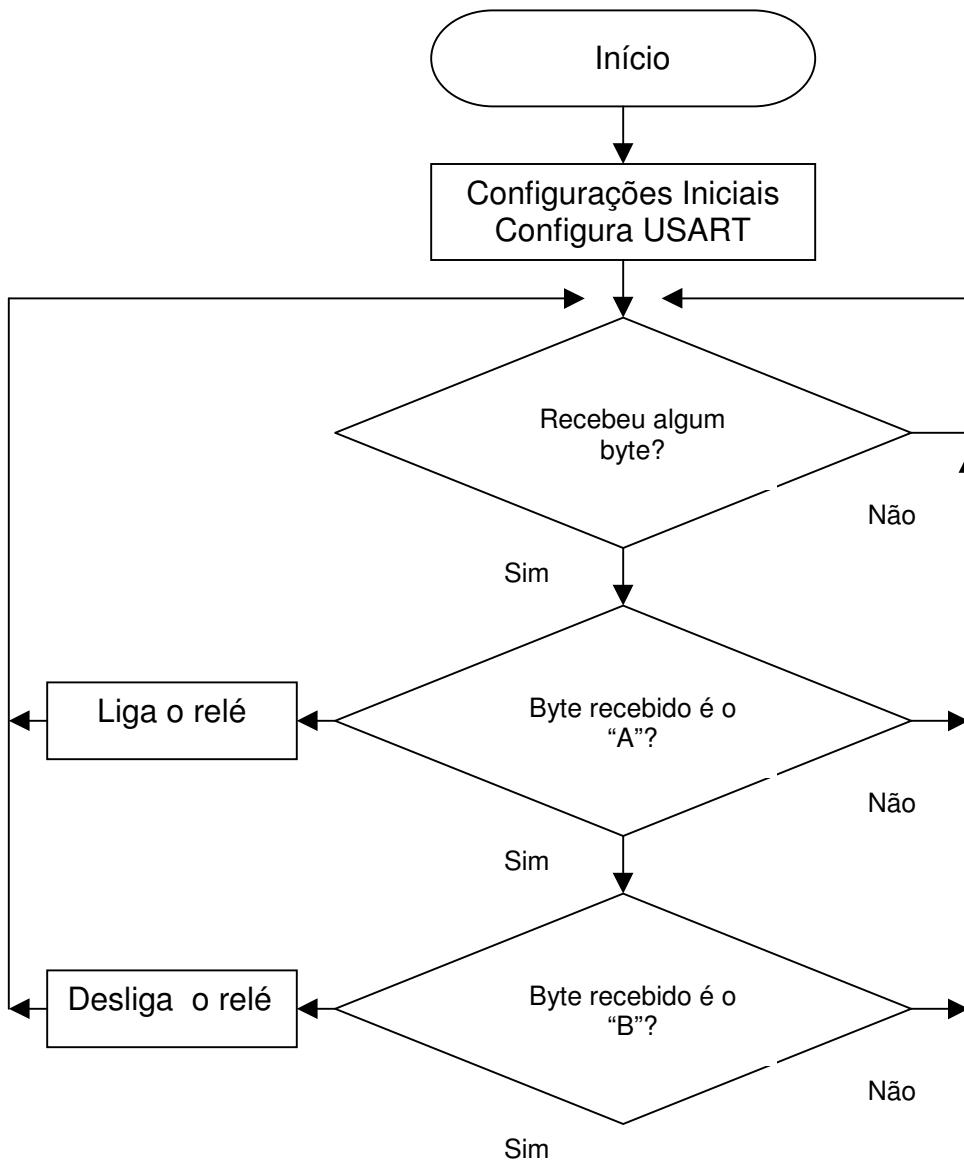


Figura 8 – Fluxograma do exemplo

Recursos de Software

Existem basicamente quatro funções disponíveis para o acesso a porta de comunicação serial do microcontrolador. Estas estão apresentadas na tabela 2.

usart_init (baud rate)
usart_read
usart_data_ready
usart_write_text (texto a escrever)

Tabela 2 – Funções de acesso a comunicação serial

Através da função `usart_init` podemos inicializar a máquina de comunicação serial com um baud rate definido pelo programa. Como no nosso caso a taxa será de 9600 bps, este comando será definido como `usart_init (9600)`. Utilizando a função `usart_read`, podemos ler o byte que foi recebido e está armazenado no buffer do microcontrolador. Já a função `usart_data_ready` permite com que saibamos se existe ou não um byte para ser feita a leitura. Sempre antes de lermos o buffer de comunicação através da função `usart_read`, precisamos saber se existe algum byte lá e através do teste da `usart_data_ready` podemos obter esta informação. Com a função `usart_write_text` podemos enviar um dado para o PC.

Software

O software completo que permitirá observar o funcionamento deste exemplo está apresentado no box 1.

```
program comunicacao
main:

    trisb.0=0
    usart_init(9600)

repete:

    if usart_data_ready=1 then
        if usart_read="A" then
            portb.0=1
        end if
        if usart_read="B" then
            portb.0=0
        end if
    end if
    goto repete

end.
```

Box 1 – Programa do exemplo

Vamos entender melhor o código apresentado. Logo destarte observamos o pino RB0 configurado como saída, através do comando `trisb.0=0`. Esta configuração é feita pois neste pino está ligado o relé. Logo na linha abaixo a `usart` é configurada para funcionar a 9600 bps através da função `usart_init`. No ponto seguinte, o programa fica preso em loop testando continuamente se o existe ou não um dado no buffer de comunicação. Ao ser detectado a presença de um byte, é verificado se o caracter é o "A" ou o "B" e caso seja algum deles será tomada a ação de acordo com o caracter.

Para que possamos enviar os dados do PC para o microcontrolador, utilizaremos o software chamado Communication Terminal que já vem no próprio mikroBASIC. Para acessar este software, vá em Tools -> Usart Terminal. Será aberta uma tela como apresentado na figura 9.

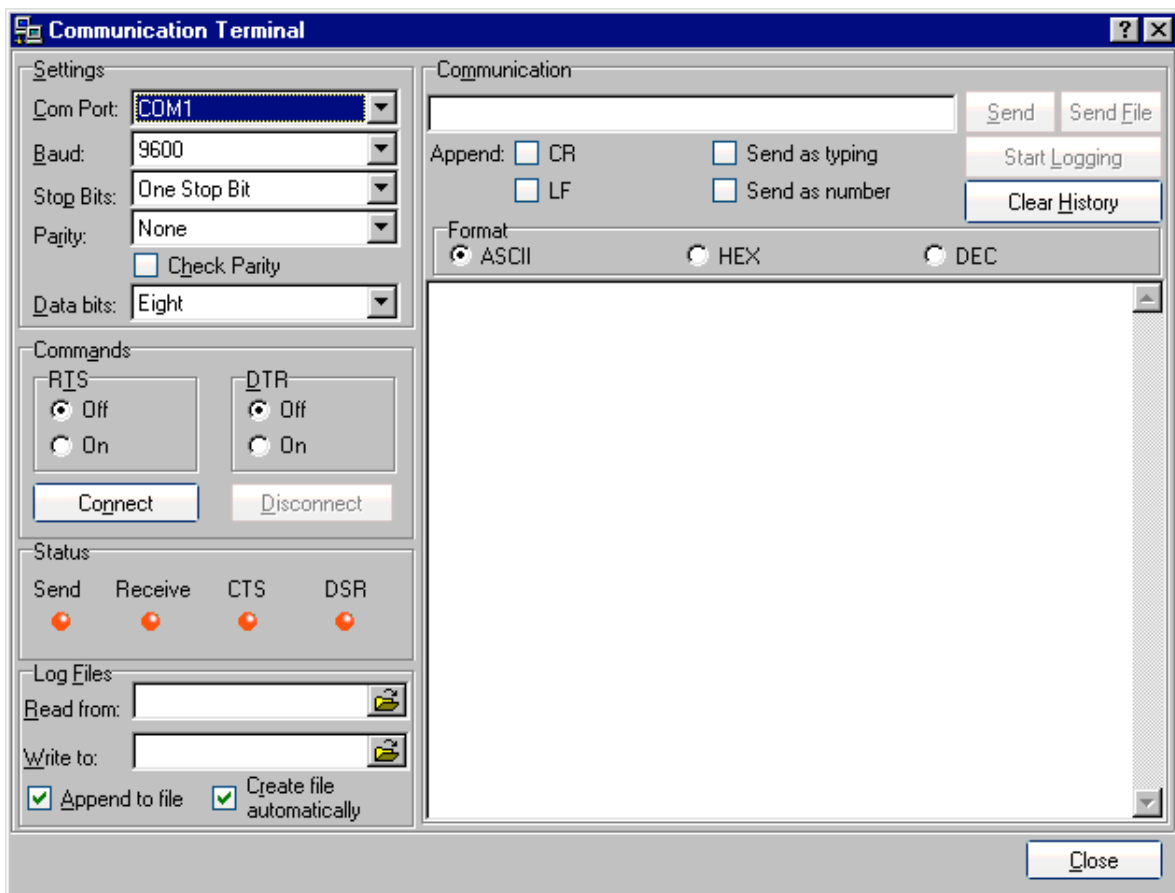


Figura 9 – Utilizando o Communication Terminal

Escolha em Com Port a porta que esteja livre em seu PC para comunicação. Em seguida, em baud mantenha a taxa de 9600 bps. Configure em Stop Bit a opção One Stop Bit e em paridade None. Utilizaremos a comunicação de 8 bits de dados, escolha desta forma esta opção em Data Bits. Muito bem, com estes parâmetros ajustados podemos iniciar a comunicação. Para isso, clique em

Connect. Neste momento a porta estará aberta e pronta para enviar dados. Na caixa Communication, digite o caracter "A" e clique em send. Neste momento, o relé deverá ser acionado. Para desativar o mesmo o processo é o mesmo, bastando apenas trocar o caracter para "B".

Conclusão

A comunicação serial é a melhor forma de permitir com que um ou mais dispositivos possam se comunicar em uma rede. Existem hoje vários barramentos seriais disponíveis, como por exemplo o CAN, RS485 e USB. Vimos neste artigo os passos para controlar um relé remotamente e podemos expandir esta idéia para uma placa que em conjunto com um software devidamente configurado no PC, permita controlar todo o funcionamento de uma casa ou indústria. Aplicações como esta, serão corriqueiras em um futuro próximo, em que a domótica cada dia mais invade as nossas casas. Comentários, dúvidas ou sugestões, deixo novamente o meu e-mail disponível para contato, que é **vitor@cerne-tec.com.br**.