



Objetivos

1. Verificar via simulação os sinais de comunicação serial assíncrona.
2. Implementar e testar via simulação códigos exemplo de comunicação via UART.

Roteiro

1

- Carregue no MPLAB o projeto “COISA.X” fornecido no pack desta prática.
- Analise o programa (trecho de inicialização e loop principal) e determine a sua funcionalidade (resultado esperado).
- Consulte o *datasheet* e confirme se o trecho de inicialização está configurando a USART para 9600 bps @ Fosc = 10 MHz, 8 databits, no-parity, 1 startbit, 1 stopbit. (Dicas: TXEN=1, BRGH=1, SPEN=1, CREN=1, SPBRG=...). Monte o projeto (*clean and build*), para assim gerar um arquivo .HEX.

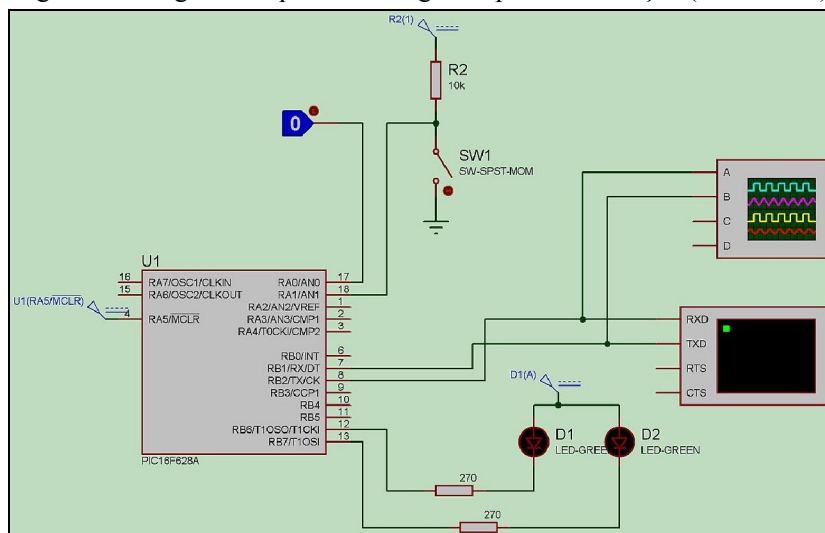
2

- Carregue no PROTEUS o projeto “TX_BAS.DSN fornecido no pack desta prática (figura 1)
- Verifique no PROTEUS (duplo clique no PIC) se o XTAL (Fosc) está ajustado em 10 MHz.
- Ainda na configuração do PIC, carregue o arquivo .HEX recém gerado pelo MPLAB.
- Coloque a simulação no PROTEUS para rodar e observe com o osciloscópio o sinal gerado no pino de TX do PIC. Verifique se o sinal gerado está de acordo com o resultado esperado.
- Meça o tempo entre transmissões (ou seja, o valor aprox. do tempo de atraso entre transmissões).
- Meça o baud rate. Dica: meça o tempo de algum bit transmitido que possa ser “isolado”.
- Altere o programa fonte para que seja enviado um caractere ‘T’ a cada 500 ms.


3

- Duplo click no VIRTUAL TERMINAL e veja as configurações (mas não altere nada).
- Ao seu critério, botão direito no VIRTUAL TERMINAL → *Echo Typed Characters*.
- Digite um ‘A’ no VIRTUAL TERMINAL e verifique no OSCILOSCÓPIO a forma de onda gerada no pino TXD do mesmo. Analise bit a bit esta forma de onda e também verifique se a taxa de bits corresponde ao valor visto nas configurações.


Figura 1 – Diagrama esquemático sugerido para a simulação (PROTEUS)



4

- Altere o programa para obter a seguinte funcionalidade:
 - **a partir do estado de um botão, mudar o caractere enviado de ‘A’ para ‘X’.** 
- Dicas: ➔ Para utilizar RA0 e RA1 como E/S, configure CMCON. (datasheet, sec. 5.1).
 ➔ Lembrar de configurar TRISA
 ➔ Super dica: “_TPININ_01.ASM”

5

- Altere o programa para obter a seguinte funcionalidade: 
 - **ao receber ‘A’ via USART, ligar o LED1;**
 - **ao receber ‘Z’ via USART, desligar o LED1;**
 - **ao receber ‘S’ via USART, ligar o LED2;**
 - **ao receber ‘X’ via USART, desligar o LED2.**

6

- Altere o programa para obter a seguinte funcionalidade:
 - **ao receber ‘K’ via USART, enviar via USART o dizer: “LEGAL !”.**

7

- Altere o programa/simulação, conforme indicado:
 - **adicione um terceiro LED (LED3);**
 - **configure o timer 0, habilite sua interrupção, e faça o LED3 piscar com 0,5 Hz;**
 - **ao receber ‘1’ via USART, alterar a frequência da piscada para 0,2 Hz;**
 - **ao receber ‘2’ via USART, alterar a frequência da piscada para 0,5 Hz;**

8

- Implemente um controle de acesso com a seguinte funcionalidade:
 - **senha de 4 dígitos (fixa), recebida via USART;**
 - **um LED deve ser ligado por 2 segundos quando a senha correta for recebida;**
 - **enquanto estiver aguardando o primeiro caractere da senha, um outro LED deve permanecer ligado.**