



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

- Crie um novo projeto no MPLAB.
- Escreva a estrutura básica de um programa.
(Dicas: “#include <P16F628A.INC>”, org 0, goto INI, org 4, retífe, INI:, MAIN:, goto MAIN).
- Monte o programa (clean and build), para assim gerar um arquivo .HEX.

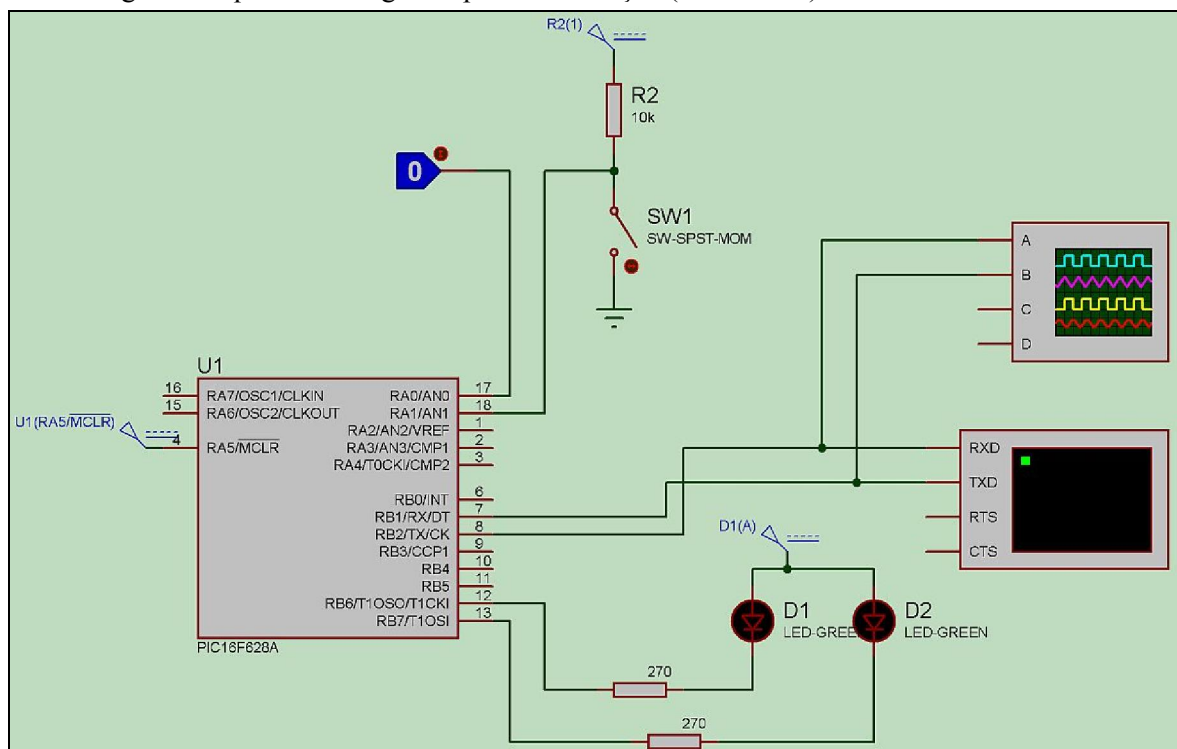
2

- Crie um novo projeto no PROTEUS, conforme indicado na figura 1. (**Componentes:** PIC16F628A, RES, LED-GREEN, SW-SPST-MOM, LOGICTOGGLE. **Instrumentos:** osciloscópio, virtual terminal).
- Carregue o arquivo .HEX e configure o XTAL do PIC para 10 MHz
- 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’ e verifique no OSCILOSCÓPIO a forma de onda gerada. Analise bit a bit esta forma de onda e também verifique se a taxa de bits corresponde ao valor visto nas configurações.


3

- Consulte o *datasheet* e configure a USART para: 9600 bps, 8 *databits*, *no-parity*, 1 *startbit*, 1 *stopbit*.
(Dicas: TXEN=1, BRGH=1, SPEN=1, CREN=1, SPBRG=?).
- Crie uma sub-rotina de atraso.
- No loop principal, inclua as seguintes tarefas: atraso de 10 ms, envio do caractere ‘A’ pela USART.
- Verifique no OSCILOSCÓPIO e no VIRTUAL TERMINAL se o PIC está gerando o sinal esperado no pino TX.


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

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.
- Dicas: ➔ BANKSEL PIR1
BTFSC PIR1,RCIF
CALL recebe
➔ O dado recebido está em RCREG.

6

- 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;

7

- Altere o programa para obter a seguinte funcionalidade:
- ao receber 'K' via USART, enviar via USART o dizer: "LEGAL !".

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.