

Universidade de São Paulo
EESC
Departamento de Engenharia Elétrica
Engenharia Elétrica - ênfase em Eletrônica

Aplicação de Microprocessadores II



Prática 1

Paulo Augusto Alves Luz Viana nºUSP 9313624
Anderson Hiroshi Siqueira nºUSP 9313197

São Carlos
2017

Conteúdo

1	Introdução	3
2	Materiais e métodos	3
2.1	Parte 1 - Comunicação Serial	3
2.2	Parte 2 - Comunicação Serial e interrupções	4
2.3	Parte 3 - Display de 7 Segmentos	4
2.4	Parte 4 - Botão e Chave	5
2.5	Parte 5 - Display LCD	6
3	Resultados	8
3.1	Parte 1	8
3.2	Parte 2	8
3.3	Parte 3	8
3.4	Parte 4	8
3.5	Parte 5	8
4	Conclusão	8
5	Bibliografia	8

Lista de Figuras

1	Protocolo RS-232	4
2	Conexão dos pinos do display	5
3	Numeração dos led's do display	5
4	Conexão do botão ao <i>port</i> do microcontrolador	6
5	Lista de comandos do LCD	7

1 Introdução

Nesta prática foram realizadas várias implementações em Assembly e em C, no microcontrolador AT89S52, baseado em 8051. Todas as implementações em Assembly foram feitas também em C, com o objetivo de se compreender e entender o funcionamento da linguagem e do hardware.

A programação em Assembly tende a ser mais complicada pela linguagem ser de muito baixo nível, mesmo em comparação a linguagem C. Porém, ela permite uma liberdade maior de acesso a memória e as portas do microcontrolador, de tal forma que também é necessário cuidado para não se invadir outras regiões do código. Em C, por outro lado, a programação se torna mais intuitiva, e visualmente mais organizada, permitindo ainda a utilização de bibliotecas. Portanto em Assembly pode-se obter uma eficiência extremamente alta em termos de memória e velocidade, porém o esforço depreendido será maior e dependendo do tamanho do seu código seja melhor usar a linguagem C, pois a diferença de performance não será tão grande.

O protocolo de comunicação serial RS232 funciona enviando-se uma sequência de bits junto com os dados, de forma serial. Há o bit de início (start bit), o stop bit (que pode ser mais de 1) que indica o fim da mensagem, e o bit de paridade que pode ou não estar presente para garantir que a mensagem foi entregue corretamente. Para que a comunicação seja feita da forma correta, é necessário que quem recebe os bits os leia na mesma velocidade que quem enviou, ou com o mesmo *baud rate*. O *baud rate* mais comum e utilizado em todas as implementações desta prática foi de 9600 bits por segundo.

Um display LCD é um painel que pode exibir caracteres ou números, tendo 1, 2 ou mais linhas de caracteres. Funciona por meio de um microcontrolador interno que pode ser configurado externamente através de seus pinos de dados (D0 à D7) e configurações (ENAB, R/W e R/S). Com as devidas combinações de valores lógicos nestes pinos pode-se escrever em sua tela, fazê-la acender, mover o cursor de escrita e ler o que está gravado nele, por exemplo.

O display de 7 segmentos é na realidade um conjunto de 7 leds (ou 8, com o ponto decimal), dispostos de forma a representar um número, e até algumas letras, normalmente utilizado para ser uma saída visual de algum dado do sistema, ou algum indicador.

2 Materiais e métodos

Kit CPU-8051-USB Placa com o microcontrolador AT89S52, da ATMEL, baseado em 8051.

Módulo PRL-LCD-1602(LCD 16x2 Black light) Painel LCD com o controlador HD44780.

Módulo PR-LED-BT(8 leds e 10 botões) Placa com pinos de entrada para escrita nos leds e saída para leitura dos botões.

Módulo PR -7SEG-4 (Display 7 segmentos de 4 dígitos multiplexado) Placa com 4 Displays de 7 segmentos de anodo comum, multiplexados por transistores ligados aos catodos dos leds.

4 Flat cables para conexão entre CPU e módulos Cabos de conexão entre as placas e o microcontrolador.

Cabo de programação USB Cabo para conexão com o PC, para programar o AT89S52.

Cabo de comunicação RS232 Cabo de comunicação com o PC.

2.1 Parte 1 - Comunicação Serial

Nesta parte foi implementada uma comunicação serial entre o microcontrolador e o computador, com mensagens sendo enviadas de um para o outro. Foi utilizado os parâmetros (9600, 8, N, 1), que significa 9600 de *baud rate*, com 8 bits de dados, sem bit de paridade e 1 stop bit. No PC os dados foram recebidos através do teclado pelo terminal do software Tera Term, que realiza a comunicação com o PC

automaticamente na forma em que estiver configurado.

Os pinos de conexão serial do AT89S52 são P3.0 (RxD) e o P3.1 (TxD), enquanto o computador receberá os dados pela sua porta USB.

A figura 1 mostra a sequência de bits enviadas para que se estabeleça a comunicação serial.

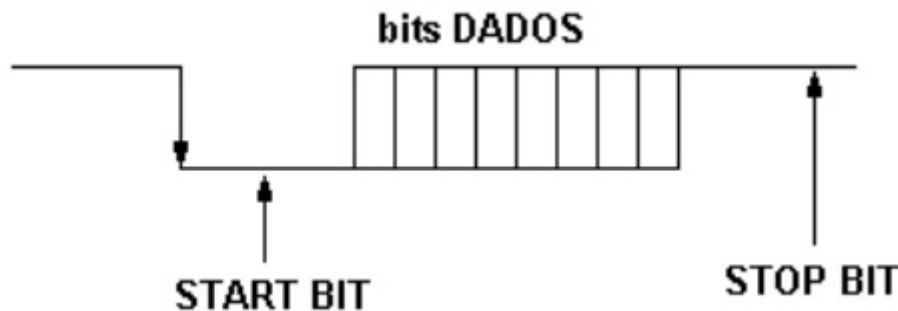


Figura 1: Protocolo RS-232

2.2 Parte 2 - Comunicação Serial e interrupções

Nesta parte os códigos da Parte 1 foram praticamente reutilizados, porém agora com o uso de interrupções para indicar o envio e a chegada de mensagens pela porta serial. O bit ES liga/desliga a interrupção serial, dependendo de seu estado lógico, e os bits RI e TI são flags que indicam que a mensagem foi enviada e recebida, respectivamente.

2.3 Parte 3 - Display de 7 Segmentos

Foi proposto a escrita de um número de 4 dígitos nos displays de 7 segmentos. Como a saída de cada pino do microcontrolador entra na base de um transistor que tem seu coletor no VCC e seu emissor no anodo de cada led, é necessário estar em tensão alta para que o led acenda. O catodo de cada led está ligado em outro pino, que deverá estar em tensão baixa, portanto, para que o led acenda.

A figura 2 mostra como os pinos do display estão conectados com o microcontrolador. CN1 e CN2 são os cabos que serão ligados à algum *port* do microcontrolador.

A figura 3 mostra o nome de cada led, bem como a qual bit do port ele está conectado.

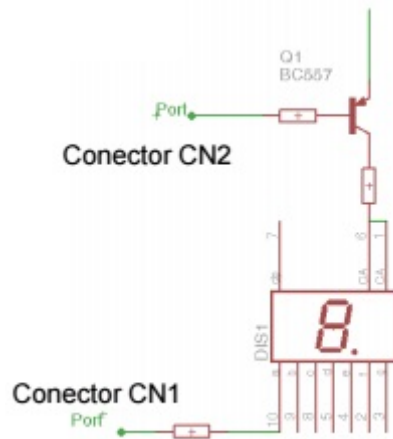


Figura 2: Conexão dos pinos do display

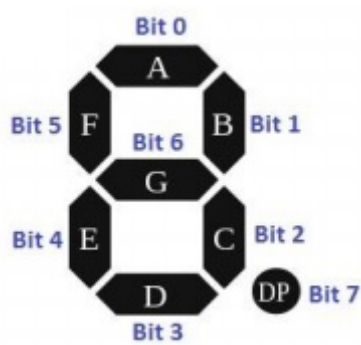


Figura 3: Numeração dos led's do display

2.4 Parte 4 - Botão e Chave

Cada *port* do microcontrolador contém um latch para cada pino. Em cada pino pode-se ler e escrever valores, porém o valor lido depende do que está pendurado no nó do pino. Pode-se então ler um valor de um botão, fazendo-se a devida montagem. É importante, porém, contornar o problema do ruído do botão, o que normalmente se faz com uma rotina de espera de tempo.

A proposta foi implementar um código que leia quais botões estão pressionados, e acende os led's correspondentes da placa. A montagem do botão está representada na figura 4.

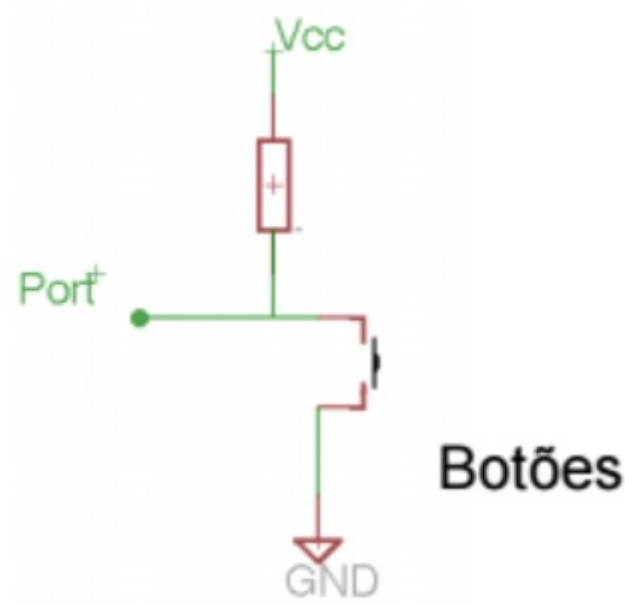


Figura 4: Conexão do botão ao *port* do microcontrolador

2.5 Parte 5 - Display LCD

O display LCD é um dispositivo um pouco mais complexo que os demais utilizados até agora. Por conter um controlador dentro de si, ele precisa receber configurações através dos pinos R/W e R/S. As rotinas INIT, POS e WRITE, presentes em quase todos os códigos como será visto, fazem essa configuração para que a escrita seja feita da forma correta.

Para se enviar cada comando, é necessário que o LCD receba uma descida de sinal no pino ENAB. A figura 5 mostra todos os comandos que se pode dar ao LCD para escrever dados, mover o cursor, ligar o display, entre outros.

INSTRUÇÃO	RS	R/W	B7	B6	B5	B4	B3	B2	B1	B0	DESCRIÇÃO e tempo de execução (uS)	t
Limpa Display	0	0	0	0	0	0	0	0	0	1	-Limpa todo o display e retorna o cursor para a primeira posição da primeira linha	1.6 mS
Home p/ Cursor	0	0	0	0	0	0	0	0	1	*	-Retorna o cursor para a 1. coluna da 1. Linha -Retorna a mensagem previamente deslocada a sua posição original	1.6 mS
Fixa o modo de funcionamento	0	0	0	0	0	0	0	1	X	S	-Estabelece o sentido de deslocamento do cursor (X=0 p/ esquerda, X=1 p/ direita) -Estabelece se a mensagem deve ou não ser deslocada com a entrada de um novo caracter (S=1 SIM, X=1 p/ direita) -Esta instrução tem efeito somente durante a leitura e escrita de dados.	40 uS
Controle do Display	0	0	0	0	0	0	1	D	C	B	-Liga (D=1) ou desliga display (D=0) -Liga(C=1) ou desliga cursor (C=0) -Cursor Piscante(B=1) se C=1	40 uS
Desloca cursor ou mensagem	0	0	0	0	0	1	C	R	*	*	-Desloca o cursor (C=0) ou a mensagem (C=1) para a Direita se (R=1) ou esquerda se (R=0) - Desloca sem alterar o conteúdo da DDRAM	40 uS
Fixa o modo de utilização do módulo LCD	0	0	0	0	1	Y	N	F	*	*	-Comunicação do módulo com 8 bits(Y=1) ou 4 bits(Y=0) -Número de linhas: 1 (N=0) e 2 ou mais (N=1) -Matriz do caracter: 5x7(F=0) ou 5x10(F=1) - Esta instrução deve ser ativada durante a inicialização	40 uS
Posiciona no endereço da CGRAM	0	0	0	1	Endereço da CGRAM						-Fixa o endereço na CGRAM para posteriormente enviar ou ler o dado (byte)	40 uS
Posiciona no endereço da DDRAM	0	0	1	Endereço da DDRAM							-Fixa o endereço na DDRAM para posteriormente enviar ou ler o dado (byte)	40 uS
Leitura do Flag Busy	0	1	B F	AC							-Lê o conteúdo do contador de endereços (AC) e o BF. O BF (bit 7) indica se a última operação foi concluída (BF=0 concluída) ou está em execução (BF=1).	0
Escreve dado na CGRAM / DDRAM	0	1	Dado a ser gravado no LCD								- Grava o byte presente nos pinos de dados no local apontado pelo contador de endereços (<i>posição do cursor</i>)	40 uS
Lê Dado na CGRAM / DDRAM	1	1	Dado lido do módulo								- Lê o byte no local apontado pelo contador de endereços (<i>posição do cursor</i>)	40 uS

Figura 5: Lista de comandos do LCD

3 Resultados

3.1 Parte 1

3.2 Parte 2

3.3 Parte 3

3.4 Parte 4

3.5 Parte 5

4 Conclusão

5 Bibliografia