

Lista de Exercícios 2

Conceitos envolvidos:

- a) Interface serial RS232
- b) Interface com Teclado Matricial
- c) Interface com Displays de 7 segmentos
- d) Interface com LCD
- e) Interface com Conversores A/D e D/A
- f) Interface usando Mapeamento de Memória
- g) Power Down, Idle, I2C, Watchdog, ISP, SPI

1) Escrever um programa em Assembly do 8051 que faça a comunicação serial RS232-C entre dois microcontroladores 89S52. A velocidade deve ser estabelecida inicialmente em 9600 Bps, 1 Stop Bit, 8 bits de dados e sem verificação de paridade. Um dos microcontroladores será o Mestre e o outro o Escravo. O Mestre envia uma "String" de caracteres ASCII na velocidade de 9600 Bps informando a Nova Velocidade, altera sua própria velocidade para o Novo Valor e o Escravo, ao receber a "String", altera sua velocidade de comunicação, devolvendo ao Mestre, na nova velocidade, uma string de reconhecimento. O cristal dos dois microcontroladores é de 11.0592 MHz. As "Strings" devem ser armazenadas na memória de programa de cada microcontrolador através da pseudo-instrução DB. O programa deve ser executado apenas uma vez.

- a. "String" a ser enviada pelo Mestre em 9600 Bps: NOVA_VEL1200
- b. "String" de reconhecimento enviada pelo Escravo em 1200 Bps: VEL_RECONHECIDA

2) Fazer um programa em Assembly do 8051 que informe a temperatura de um forno. O programa deve enviar inicialmente um 'String' em ASCII armazenado na área de memória de programa a partir do endereço "TAB", para um monitor de vídeo serial RS232C (600 bps, sem paridade, 8 bits de dados, 1 stop bit), até encontrar o código ASCII do caractere "\$" que não deve ser transferido. Após a mensagem ser mostrada, o programa deve enviar o valor da temperatura que está armazenado em hexadecimal no registrador R0 e parar. O cristal do micro é de 11.0592 Mhz.

TAB: DB 'TEMPERATURA DO FORNO =', '\$'

3) Uma determinada máquina possui dois dispositivos que geram pulsos não sincronizados entre si. Cada fonte de pulsos está ligada nas Interrupções externas Int0 e Int1 de um microcontrolador 8051. Desenvolver um programa em Assembly que conte estes pulsos e envie a contagem para as Portas, da seguinte maneira:

- a) Os pulsos ligados na Int0 são contados a cada decada de borda e o total enviado para a Porta P1.
- b) Os pulsos ligados na Int1 são contados em nível zero e enviados para a Porta P2.
- c) Os pulsos ligados na Int0 são prioritários, ou seja, se ocorrer uma Int1 durante o atendimento de uma Int0, seu atendimento somente será realizado no final.
- d) Cada vez que uma contagem (Int0 ou Int1) "estourar" outro contador, cuja saída é a Porta P0, deve ser incrementado. O valor da Porta P0 deve ser também enviado para um monitor de vídeo RS232 operando na velocidade 4800,N,8,1 e em uma só posição, ou seja, o contador ocupa sempre os mesmos locais na tela.
- e) Quando o contador da Porta P0 "estourar" o sistema deve re-iniciar e as saídas dos Contadores (P0, P1, P2) devem ser zeradas. O sistema deve enviar para o monitor de vídeo a informação "CONTAGEM RE-INICIADA" e reposicionar os dígitos ASCII na tela logo abaixo da mensagem para continuar mostrando a contagem a partir de 00.

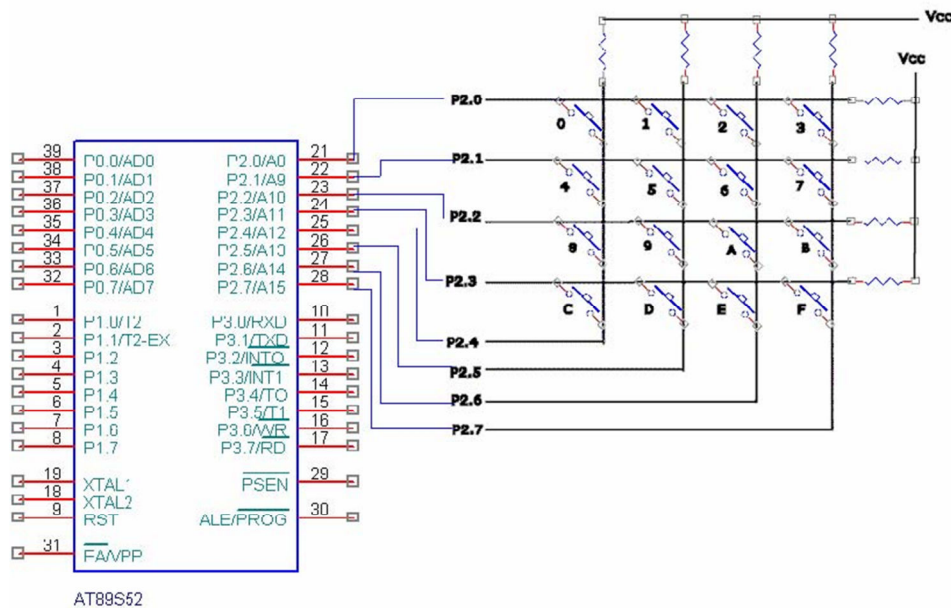
- 4) Dois monitores de vídeo estão conectados a um microcontrolador 8051 sendo selecionados através do pino de porta P1.0. Se P1.0 = 0 a comunicação serial RS232 se faz com o monitor de vídeo 1 na velocidade 9600,N,8,1. Se P1.0 = 1 a comunicação via SBUF é com o monitor de vídeo 2 na velocidade 4800,N,8,1.

Desenvolver um programa em Assembly do 8051 que ordene crescentemente uma sequencia de números inteiros de 8 bits, armazenada na memória de programa e envie alternadamente seus valores em ASCII para cada um dos monitores, começando do primeiro número (o menor) para o monitor 1, segundo para o monitor 2 e assim por diante até encontrar o último número da sequencia que obrigatoriamente deve ser FF, parando o programa.

Sequencia na memória de Programa:

SEQ: DB 89h,56h,0d3h,4eh,0a7h,23h,0ffh,00h,99h,7fh,12h,0ceh,0bch,02h,0fdh,66h

- 5) Desenvolver um programa em Assembly do 8051 que envie um caractere ASCII "*" para um monitor de vídeo variando a sua velocidade de comunicação a cada 5 segundos. A velocidade deve começar em 75,N,8,1 e avançar em múltiplos de 2 até 9600,N,8,1. Depois de 5 segundos da velocidade máxima, retornar para a velocidade mínima e recomeçar.



- 6) Escrever um programa em Assembly do 8051 que comande dois dispositivos conectados à Porta P1 que controlam uma fechadura para acesso seguro a um ambiente industrial. Somente a senha C42F digitada no teclado matricial, conforme esquema, abre a fechadura através do dispositivo D1 conectado no pino P1.0 quando este bit for setado. Somente a senha F24C aciona o dispositivo D2, conectado ao pino P1.1 que fecha a mesma fechadura quando este bit for setado. Qualquer outra senha digitada zera os pinos P1.0 e P1.1, aciona o alarme A1 conectado ao pino P3.7 setando este bit e aguarda a senha de desbloqueio (369C) que desliga o alarme e permite ao programa voltar ao controle da fechadura.

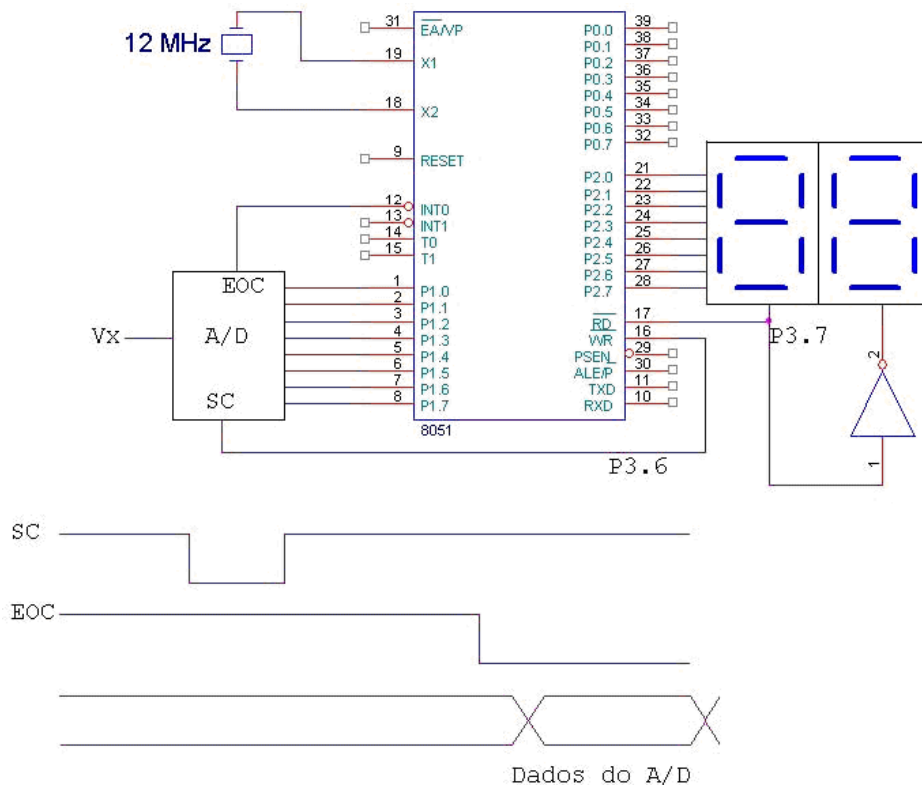
7) Para o esquema de um Voltímetro com fundo de escala de 15 volts e com resolução de 1 volt construído com o Microcontrolador 8051, considerar os seguintes dados :

- O Conversor A/D tem tempo de conversão de 100 us (microsegundos) isto é, este é o tempo decorrido entre a subida de borda do pulso SC (Start Conversion) a ser gerado pelo microcontrolador no pino P3.6 e a descida de borda gerada pelo A/D através do sinal EOC (End of Conversion) indicando que os dados estão disponíveis para serem lidos.
- A duração do pulso de SC (Start Conversion) deve ser de no mínimo 50 us (microsegundos).
- Os dois Displays de 7 segmentos são do tipo catodo comum e multiplexados através do pino P3.7 que comanda qual Display será aceso no tempo.
- O valor a ser aceso no Display deve ser disponibilizado na Porta P2 na ordem:



O Pino P2.7 não está sendo usado e cada Display mostrará valores de 0 a 9.

Escrever o Programa Principal que deve converter os dados armazenados no Buffer de Display (endereços da RAM Interna 30h = LSB e 31h = MSB) para 7 segmentos e enviá-los para os Displays Multiplexados, permanecendo continuamente em loop. Programar a Interrupção INT0 para responder ao Conversor A/D. Os valores armazenados no Buffer de Display vão de 00 a 09 para o endereço 30h e de 00 a 01 para o endereço 31h .



Escrever a sub-rotina de atendimento de Interrupção do Conversor A/D que deverá ler

o valor Digital equivalente da Tensão V_x , de acordo com o diagrama de tempos fornecido, e armazená-lo nas posições 30h e 31h da RAM interna (Buffer dos Displays) da seguinte forma:

Valor em P1 – Displays – Buffer dos Displays

De 00 a 0F → 00 volts → 31h = 00 e 30h = 00
De 10 a 1F → 01 volts → 31h = 00 e 30h = 01
De 20 a 2F → 02 volts → 31h = 00 e 30h = 02
De 30 a 3F → 03 volts → 31h = 00 e 30h = 03
De 40 a 4F → 04 volts → 31h = 00 e 30h = 04
De 50 a 5F → 05 volts → 31h = 00 e 30h = 05
De 60 a 6F → 06 volts → 31h = 00 e 30h = 06
De 70 a 7F → 07 volts → 31h = 00 e 30h = 07
De 80 a 8F → 08 volts → 31h = 00 e 30h = 08
De 90 a 9F → 09 volts → 31h = 00 e 30h = 09
De A0 a AF → 10 volts → 31h = 01 e 30h = 00
De B0 a BF → 11 volts → 31h = 01 e 30h = 01
De C0 a CF → 12 volts → 31h = 01 e 30h = 02
De D0 a DF → 12 volts → 31h = 01 e 30h = 03
De E0 a EF → 14 volts → 31h = 01 e 30h = 04
De F0 a FF → 15 volts → 31h = 01 e 30h = 05

Na figura abaixo está o circuito de um sistema a microprocessador baseado no Microcontrolador AT89S52 da Atmel. As questões 8 e 9 referem-se a este circuito.

8) Desenvolver um programa em Assembly que leia um valor analógico no intervalo 0 a 10 v colocado em uma das entradas IN_y do conversor AD ADC0808. (y é um número de 0 a 7)

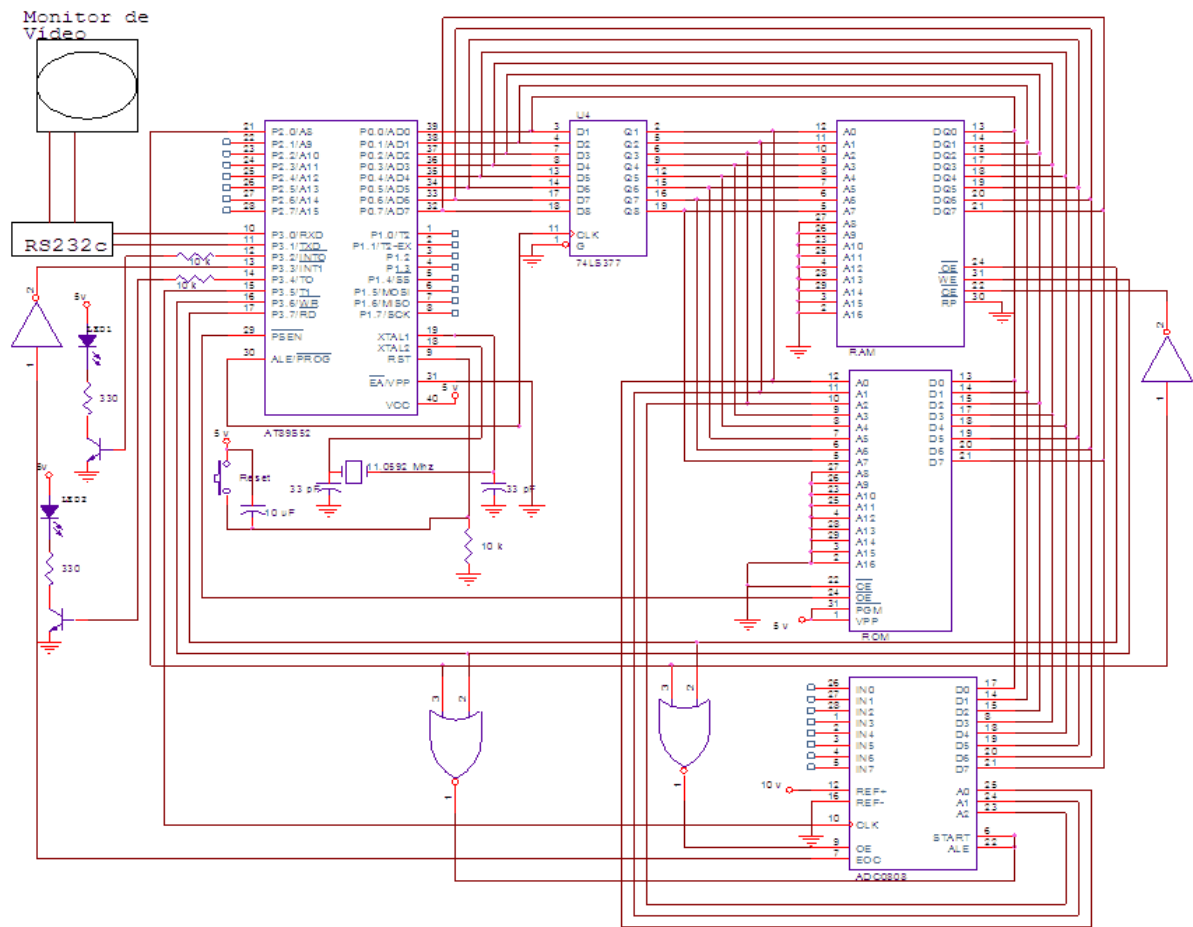
Os valores gerados pelo conversor equivalem à seguinte tabela:

0v --- 00
1v --- 17h
2v --- 2Eh
3v --- 45h
4v --- 5Ch
5v --- 73h
6v --- 8Ah
7v --- A1h
8v --- B8h
9v --- CFh
10v --- E6h

O valor da tensão lida no conversor AD deve ser enviado para o Monitor de Vídeo a uma taxa de 9600 BPS, sem paridade e 1 Stop Bit, acompanhado da seguinte informação:

VALOR ANALOGICO DA TENSAO NO SENSOR y = xx volts

onde xx é um número de 00 a 10 equivalente à tensão na entrada INy e y é um número



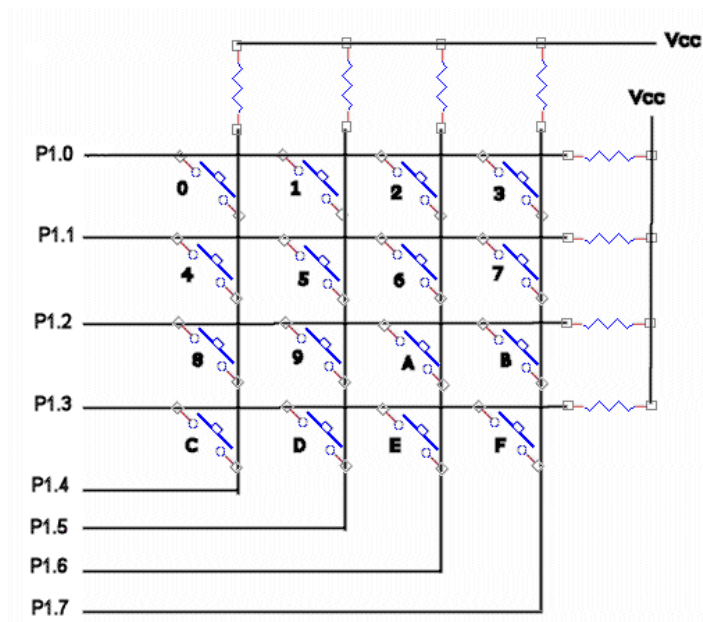
de 0 a 7 equivalente ao sensor ligado a ela. O programa deve enviar oito valores acompanhados de suas respectivas informações como explicado acima, e ficar em loop para que quando ocorra uma mudança nos sensores, esta mudança apareça diretamente no Monitor de Vídeo.

O Clock do conversor deve ser gerado pelo Timer 0 e deve ser de aproximadamente 50 KHz.

Exemplo da tela do Monitor de Vídeo:

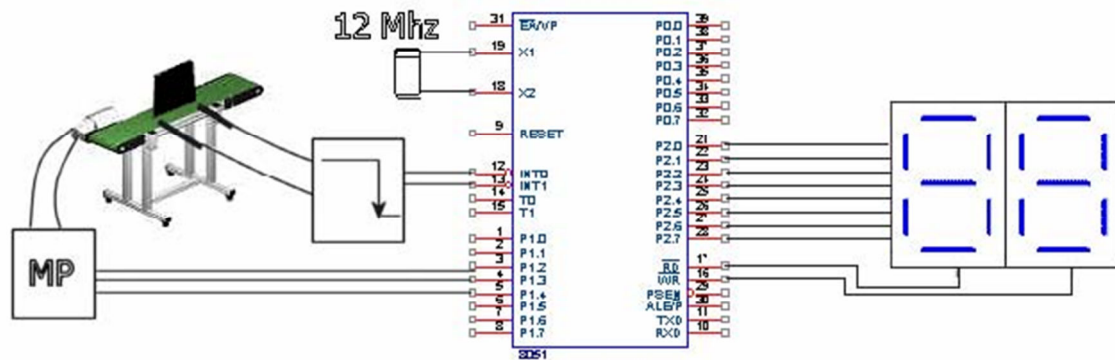
- VALOR ANALOGICO DA TENSAO NO SENSOR 0 = 05 volts
- VALOR ANALOGICO DA TENSAO NO SENSOR 1 = 09 volts
- VALOR ANALOGICO DA TENSAO NO SENSOR 2 = 00 volts
- VALOR ANALOGICO DA TENSAO NO SENSOR 3 = 03 volts
- VALOR ANALOGICO DA TENSAO NO SENSOR 4 = 10 volts
- VALOR ANALOGICO DA TENSAO NO SENSOR 5 = 01 volts
- VALOR ANALOGICO DA TENSAO NO SENSOR 6 = 04 volts
- VALOR ANALOGICO DA TENSAO NO SENSOR 7 = 08 volts

9) Conectar um teclado matricial conforme a figura, na Porta P1 do Microcontrolador.



Escrever um programa em Assembly que permita a um usuário digitar uma senha de 4 dígitos no teclado. Se a senha digitada for 4AD3 em hexadecimal, o programa deve acender o LED1 (P3.2) e preencher a RAM externa tal que o conteúdo de cada posição seja igual à parte menos significativa do endereço. Se a senha for CABE em hexadecimal a RAM externa deve ser totalmente zerada e o LED2 (P3.4) aceso. Qualquer outra senha que for digitada diferente das duas anteriores, o programa deve preencher a RAM externa com FFh, piscar os dois leds de maneira alternada (enquanto um está aceso o outro está apagado e vice-versa) em uma frequência aproximada de 10 Hz e não permitir a entrada de nenhuma outra senha. Obs: Os Leds inicialmente devem ficar apagados.

10) Um Microcontrolador baseado no 8051 verifica o tamanho de cada peça sobre uma esteira transportadora conforme figura abaixo. A esteira é movimentada através de um Motor de Passo (MP) sendo que cada passo no motor corresponde a 1 mm (milímetro) de deslocamento linear da esteira. Os pinos de comando do MP são: P1.2 = clock, P1.3 (dir) = 1 move a esteira da esquerda para a direita e P1.4 = pára o MP se igual a zero. Na figura, a peça sobre a esteira desloca-se da esquerda para a direita e é colocada na mesma antes do primeiro sensor. O primeiro sensor, conectado em INT1, deve ser usado para medir a largura da peça em milímetros (de 1 a 9 mm) que deverá ser mostrada no Display de 7 segmentos (catodo comum) mais à direita, somente após a mesma passar completamente pelo segundo sensor ligado em INT0. Este segundo sensor deve contar o número de peças medidas (de 1 a 9) e o valor mostrado no primeiro Display (catodo comum) à esquerda. Os dois sensores devem ativar as interrupções equivalentes na descida de borda quando o objeto sai da frente do sensor. Quando o objeto está na frente do sensor o valor nos bits de porta equivalentes a INT0 = P3.2 e INT1 = P3.3 é igual a um.



- a) Escrever o Programa Principal em Assembly que continuamente:
- Comande o Motor de passo a uma velocidade de aproximadamente 200 Hz. O motor não deve ser parado movendo a esteira continuamente.
 - Avalie os sensores e meça a largura da peça através do sensor conectado em INT1.
- ATENÇÃO:** Os Buffers somente serão atualizados pela Interrupção INT0.

b) Escreva as duas Sub-rotinas de Atendimento de Interrupção em Assembly da seguinte forma:

Atualize os dois Displays. O LSB (Display à direita) com o tamanho da peça (de 1 a 9 mm) e o MSB (Display à esquerda) com o número de cada peça (1 a 9). Os dois valores devem aparecer simultaneamente (multiplexados pelos pinos P3.6 para o LSB e P3.7 para o MSB). Quando nenhuma peça tiver sido medida deve aparecer 00 nos dois Displays. Utilizar como Buffer de Display os endereços 30h = LSB e 31h = MSB da RAM Interna. O valor a ser aceso nos Displays deve ser disponibilizado na Porta P2 na ordem:



Onde P2.7 = 0 e P2.0 = g

A sub-rotina da INT0 deve atualizar os dois valores (de 1 a 9) nos Buffers (30h = LSB largura da peça e 31h = MSB número da peça).

As duas sub-rotinas devem retornar ao Programa Principal para contínua atualização dos Displays.

A sub-rotina da INT1 deve apenas parar a medida da peça, consolidar o valor em um Registrador intermediário e não deve atualizar o Buffer 30h o que deverá ser feito apenas no final do processo (quando a peça terminar de passar pelo segundo sensor solicitando a interrupção INT0).

- Fornecer o programa em Assembly
- Os parâmetros de programação dos tempos envolvidos