

----- TESTE (SEM CONSULTA) -----

I. Questões Gerais

- Qual a diferença entre interrupções por temporização ou interrupções por acontecimentos?
- Qual o tipo de interface/comunicação que é usado pelos módulos Bluetooth (BT)?
- Com base na figura seguinte comente a seguinte afirmação, “a figura seguinte representa o funcionamento em **Modo 1** do **Timer 1** do μC 8051”.

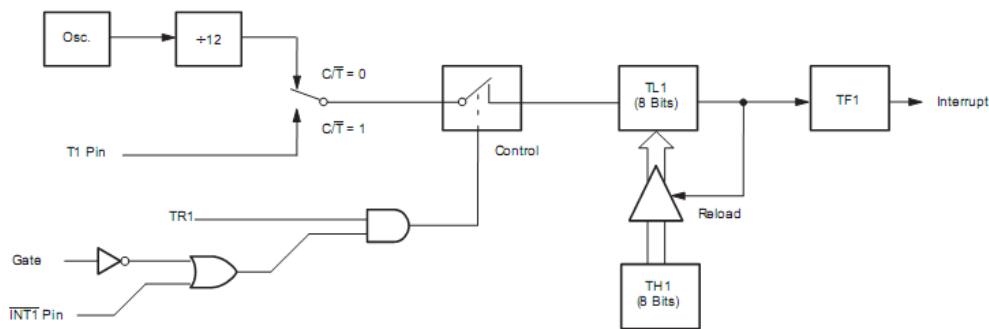


Fig.1

II. Endereçamento

- Um micro Motorola da família 68000 (antigos processadores da Apple) com **23** linhas de endereços, quantos endereços é capaz de endereçar?
- Uma RAM de **128KB** selecciona uma gama de endereços desde 010000H até ----- H.
- Diga qual o endereço e espaço de endereçamento ocupado pelo dispositivo **DEVICE#6** do circuito de descodificação com 2 x descodificadores 3:8 mostrado na figura seguinte.

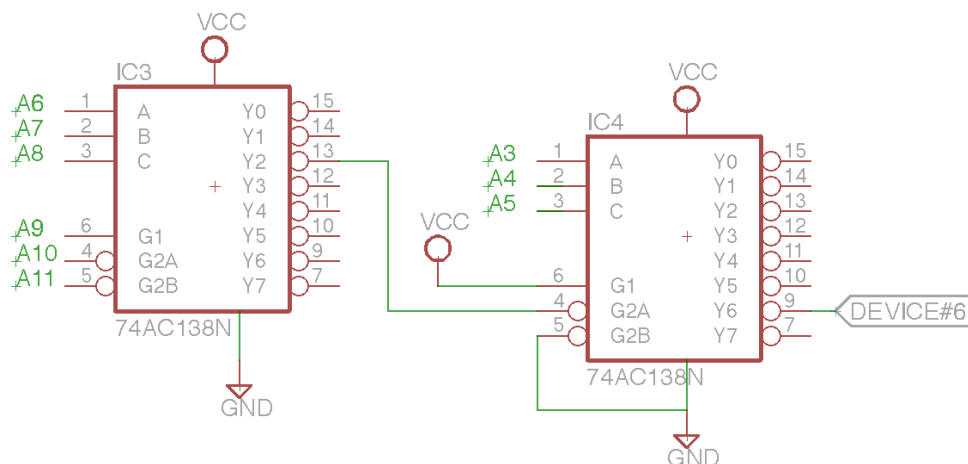


Fig.2

III. Interface de Hardware:

Fig.3



- (a) Identifique cada um dos interfaces representados na fig.3 e dê exemplos práticos de aplicação?
- (b) Qual a diferença entre motor DC e um servo-motor?
- (c) O dispositivo nº3 da figura, pode ser usado tanto como entrada ou como saída. Explique esta afirmação.
- (d) Que procedimento deveria usar para ler sinais digitais entre 0-10 Volts usando um microcontrolador tradicional de 8bits cujas entradas foram especificadas para lidar com sinais entre 0-5V?

IV. Software:

A palavra chave num sistema embebido de tempo real é o “**processamento determinístico**”, isto é, precisamos ser capazes de garantir que uma determinada atividade será sempre concluído num prazo especificado, se o processamento não satisfaz esta especificação, a aplicação torna-se inútil. O código seguinte tem 2 linhas com potenciais problemas. Identifique essas linhas e explique qual o problema que esse tipo de codificação pode provocar.

```
bit SWITCH_Get_Input(const unsigned char DEBOUNCE_PERIOD)
{
    bit Return_value = SWITCH_NOT_PRESSED;

    if (Switch_pin == 0)
    {
        // Switch is pressed - Debounce - just wait...
        DELAY_LOOP_Wait(DEBOUNCE_PERIOD);
        // Check switch again
        if (Switch_pin == 0)
        {
            // Wait until the switch is released.
            while (Switch_pin == 0);
            Return_value = SWITCH_PRESSED;
        }
    }
    // Now (finally) return switch value
    return Return_value;
}
```

V – Mini-projeto – “Cafeteria iTech baseada em microcontrolador”

Esta unidade económica é uma solução ideal para pequenos hotéis. O cliente pode fazer um pedido do item pressionando a *tecla* correspondente ao item que ele ou ela gostaria de pedir. A outra metade da unidade é o *visor* LCD, que é mantida na cozinha. O item será exibido no LCD e, posteriormente, é servido ao cliente. Uma *interrupção* será ativada quando o cliente quer pedir a conta ou para colocar um item que não está no menu ou em caso de quaisquer outros eventos. A cafeteria está equipada na cozinha com um *sensor de temperatura*, que quando a temperatura sobe acima do valor pré-determinado definido pelo utilizador, é detectado um incêndio, e a unidade emite um alarme. Projete a aplicação descrita usando o microcontrolador de sua preferência.



Questões:

- (a) Qual o micro que escolheu para implementar esta aplicação e porque ?
- (b) Faça um esquemático pormenorizado do sistema, mostrando e justificando claramente as ligações aos portos/bits que vai usar. Valoriza-se a estratégia que acrescentar novas e úteis funcionalidades.
- (c) Com base no texto “O cliente pode fazer um pedido do item pressionando a *tecla* correspondente ao item que ele ou ela gostaria de pedir. A outra metade da unidade é o *visor* LCD, que é mantida na cozinha. O item será exibido no LCD ..”, codifique o procedimentos para implementar esta tarefa. Considere a existência da biblioteca **#include <lcd.h>** que inclui as funções pré-definidas **lcd.begin(cols, rows); lcd.print(string); lcd.write(onechar) ; lcd.clear(); lcd.setCursor(col, row); lcd.clear();**Nota: Não precisa codificar estas funções, só precisa de usar as funções.
- (d) Com base no texto “Uma *interrupção* será ativada quando o cliente quer pedir a conta ou para colocar um item que não está no menu ou em caso de quaisquer outros eventos”, diga a que interrupção se refere e codifique a respetiva rotina ISR.
- (e) Qual o sensor de temperatura que poderia usar? Pode liga-lo diretamente aos portos do micro que escolheu? Porquê ?
- (f) Implemente a tarefa periódica **Detect_Fire()** que DETETA ou NÃO a presença de fogo e faz acionar os respetivos procedimentos de segurança.

Boa Sorte !!!

Necessário aumentar a eficiência de um sistema embebido

- Porque o código (software) tem erros
- Porque o mundo real pode não funcionar da maneira que esperamos e como o sistema foi projectado
- Os componentes também têm falhas

Soluções propostas

4 Mecanismos para o aumento de eficiência (maior robustez do sistema)

- Watch-Dog Timer
- Monitorização da pilha (SP)
 - Monitorização da alimentação (Black-out, Brown-out, drop-out)
 - WDT
 - 4 Detectar se o software não está a funcionar correctamente.
 - 4 Existe um sinal “batimento” que é gerado por uma tarefa que efectua RESET ao contador de Watch- Dog
 - 4 Quando esse “batimento” não existe, significa que o sistema está “doente”, e um RESET deve ser gerado (coloca o sistema num estado pré definido)

WDT

4 O WDT deve ser implementado por um relógio preciso e fidedigno – de modo a garantir o funcionamento correcto do mesmo

DATA: 18-JUN-2015

DURAÇÃO: 2.15 HORAS

5 O programa nunca pode desligar o WDT inadvertidamente

5 Saber escolher o período do WDT

(Aumento da eficiência.pdf)



Source: © iStockphoto.com/Alexey Dudoladov

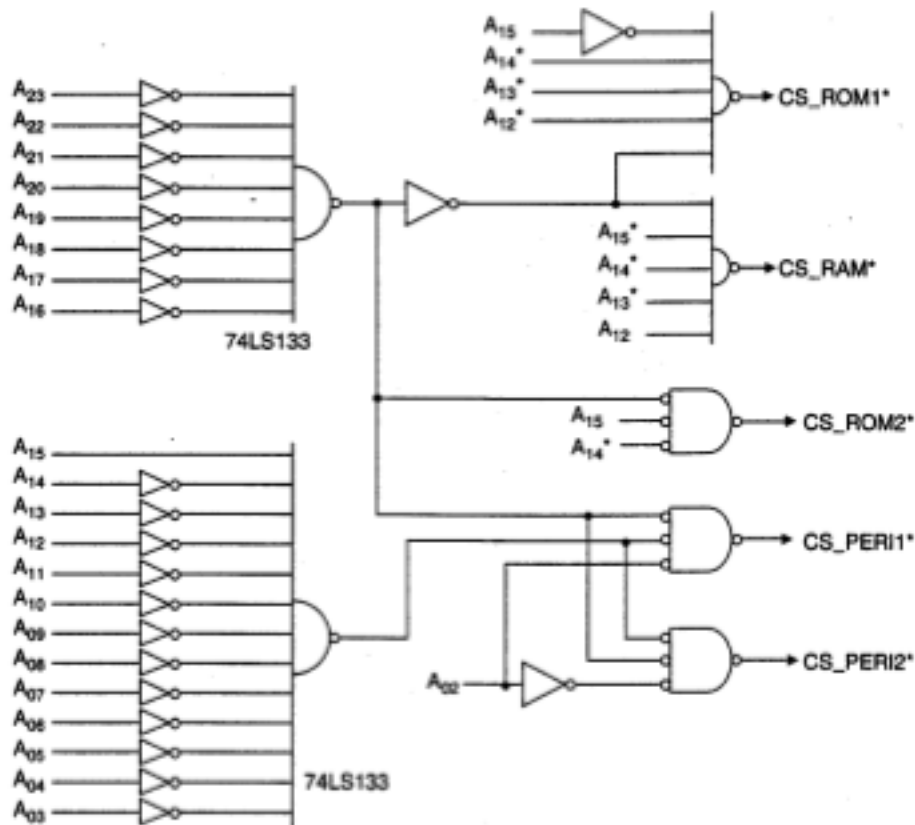
● Estratégias de Decodificação de Endereço

○ Decodificação Total - Exemplo 2: Conectar a um 68000 os seguintes componentes:

- 10 K Words ROM (2K Words + 8 K Words) - ROM1(2K) ROM2(8K)
- 2 K Words RAM
- 2 Words Periférico 1 - PERI1
- 2 Words Periférico 2 - PERI2

○ Decodificação Total - Exemplo 2:

Device	A ₂₃	A ₂₂	A ₂₁	A ₂₀ ...A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁
ROM1	0	0	0	0...0	0	0	0	X	X	X	X	X	X	X	X	X	X	X
RAM	0	0	0	0...0	0	0	1	X	X	X	X	X	X	X	X	X	X	X
ROM2	0	0	0	0...0	1	X	X	X	X	X	X	X	X	X	X	X	X	X
PERI1	0	0	0	0...1	0	0	0	0	0	0	0	0	0	0	0	0	0	X
PERI2	0	0	0	0...1	0	0	0	0	0	0	0	0	0	0	0	0	1	X



--

Dividing Voltages

Voltage dividing is probably the easiest transformation you can perform on sensor outputs to alter the value being connected to a microcontroller or other circuit.

The mathematical equivalent of what you are trying to achieve when dividing voltages is a simple division. For instance, say you have a sensor that outputs 0-100V and you want to convert this to 0-

DATA: 18-JUN-2015

DURAÇÃO: 2.15 HORAS

5V for interface to the A/D input on your Arduino. The goal would be to create a 20:1 ratio of voltage which means dividing the original sensor output voltage by a factor of 20

--

Aplicação 3: Cafeteria Food Alert / Menu Card baseado em microcontrolador

Esta unidade económica é uma solução ideal para pequenos hotéis. O cliente pode fazer um pedido do item pressionando a tecla correspondente ao item que ele ou ela gostaria de pedir. A outra metade da unidade é o visor LCD, que é mantida na cozinha. O item será exibido no LCD e, posteriormente, serviu para o cliente. Uma interrupção será invocado no caso de o cliente quer projeto de lei ou para colocar um item que não está lá no menu ou em caso de quaisquer outros eventos.