

Trabalho de Programação 2

Parte 1

Processador CESAR16i

1. Descrição Geral

Dando continuidade ao trabalho desenvolvido para o RAMSES, você deverá desenvolver o controle da movimentação da sonda. Esse controle será feito usando o processador CESAR16i e terá três partes:

- Software que fornece as informações sobre a posição e movimentação da sonda, fornecido pelo professor;
- Programação das interrupções e tratadores de interrupção, que corresponde à primeira parte do trabalho e está descrita nessa especificação;
- Programa Principal, que corresponde à segunda parte do trabalho e será motivo da próxima especificação.

Nessa especificação está descrita a primeira parte do trabalho.

2. Implementação

Nessa primeira parte do trabalho você deverá implementar, usando o mecanismo de interrupção disponível no CESAR16i, a atualização das informações de posição e movimentação da *Sonda*. Você deverá desenvolver os tratadores de interrupção de teclado e de tempo (*timer*).

O **tratador de interrupção de teclado** será usado para capturar os toques no teclado que serão usados para controlar a sonda.

O **tratador de interrupção de tempo** será usado para ler o GPS e, a partir dessa informação, calcular a distância até o destino. Nesse tratador também deve ser acionado o software de atualização de posição e movimentação da sonda (“Rotina de atualização” – vide item 3).

2.1. Tratador de Interrupção de Teclado

Sempre que o operador digitar uma tecla será gerada uma interrupção de teclado. Você deve implementar o tratador de interrupção capaz de ler a tecla que gerou a interrupção e processá-la adequadamente.

Apenas as teclas “O”, “W” e “S” (e suas correspondentes minúsculas) serão interpretadas. As restantes devem ser ignoradas.

A função de cada uma das teclas é a seguinte:

- Tecla “O”: liga/desliga o motor da sonda;
- Tecla “W”: acelera a sonda;
- Tecla “S”: desacelera a sonda

Como se pode perceber, todas as teclas atuam sobre a movimentação da sonda. A forma como isso é feito está descrito no item 2.3.

2.2. Tratador de Interrupção de Tempo

Você deve implementar o tratador de interrupção de tempo que terá como responsabilidade as seguintes tarefas:

- Chamar, a cada 100ms, a “Rotina de atualização” das informações de posição e movimentação da sonda. Para isso, basta chamar o endereço 8200₁₆ a cada 100ms, usando a chamada “**JSR R7,H8200**”;
- Transformar as informações fornecidas pelo GPS da sonda, que é uma informação de tempo, em uma medida de distância entre a posição atual da sonda e seu destino. A forma como fazer essa transformação está descrita na variável “GPS” do item 5;
- Gerar um sinal de temporização para uso do Programa Principal.

Além das tarefas previstas acima, você também poderá programar outras funções que julgar serem necessárias.

2.3. Movimentação da Sonda

O controle de movimentação da sonda é feito através das variáveis “ON” e “AR”.

Para ligar o motor da sonda, deve-se colocar um valor diferente de zero na variável “ON”. Para desligar o motor, deve-se colocar 0 (zero) na variável “ON”.

Para acelerar (aumentar a velocidade) da sonda deve-se colocar um valor diferente de zero na variável “AR”. Para reduzir (diminuir a velocidade) da sonda deve-se colocar um valor 0 (zero) na variável “AR”.

2.4. Sinal de Temporização

O tratador de interrupção de tempo deve gerar um sinal de temporização de 1 (um) segundo para ser usado pelo Programa Principal. Para isso, a cada 1 (um) segundo, o tratador deve escrever um valor diferente de zero na variável “TICK”.

A forma como o Programa Principal utilizará essa informação estará descrita na especificação da parte 2 do trabalho.

3. Divisão do espaço de endereçamento (alocação de memória)

Para fins do trabalho, o espaço de endereçamento do CESAR16i (64Kbytes) será dividido em partes. Essa divisão visa organizar as informações na memória e permitir a correção individual de cada uma das partes do trabalho. O arquivo “REF.CED” (vide item 3.1) fornecido pelo professor define essas áreas de memória usando a diretiva ORG, conforme descrito na tabela abaixo:

Faixa de Endereços (em hexadecimal)	Descrição
0000 ₁₆ até 7FFF ₁₆	Programa Principal , que deve iniciar em 0000 ₁₆ , e área de <i>stack</i> . Essa área será preenchida na segunda parte do trabalho.
8000 ₁₆ até 80FF ₁₆	Variáveis de comunicação entre o Programa Principal e a Interrupção, conforme especificado.
8100 ₁₆ até 81FF ₁₆	Rotina de inicialização de periféricos e interrupção, que deve iniciar em 8100 ₁₆ .
8200 ₁₆ até 83FF ₁₆	Rotina de atualização da posição e movimentação da sonda, que deve iniciar 8200 ₁₆ . Essa rotina deve ser chamada a cada 100ms.
8400 ₁₆ até FFFF ₁₆	Tratadores de Interrupção . As rotinas dos tratadores de interrupção podem ser colocadas em qualquer endereço dessa área de memória. Entretanto, deve-se programar adequadamente o vetor de interrupção.
FF80 ₁₆ até FFFF ₁₆	Área onde estão mapeados os periféricos : teclado, visor e timer.

3.1. Arquivo “REF.CED”

Está sendo fornecido um arquivo chamado “REF.CED” que, além da definição das áreas descritas na tabela anterior, contém a “Rotina de atualização” da situação da *Sonda*.

A sua implementação das duas partes desse trabalho deverão ser feitas sobre esse arquivo. O motivo disso é que a “Rotina de atualização”, que fornece as informações do estado da *Sonda*, deve ser chamada a cada 100ms. Essa rotina processa as variáveis de controle (ON e AR) e fornece como resultado o estado da Sonda (variáveis CARGA, GPS e VELOCIDADE).

O seu tratador de interrupção ser colocado na área prevista e deverá ser programado adequadamente, de forma que a “Rotina de atualização” seja chamada a cada 100ms.

O arquivo “REF.CED” com a sua implementação deverá ser entregue.

4. Sua tarefa nessa primeira parte do trabalho

Nessa parte do trabalho você deve programar o CESAR16i para que as interrupções possam ser acionadas e implementar os tratadores de interrupção. Para isso, você deve inicializar as variáveis desses tratadores, configurar os

periféricos (incluindo a programação da resolução do timer), inicializar as interrupções e o programar o tratamento das variáveis compartilhadas (ver item 5).

O valor programado no timer pode ser escolhido segundo suas necessidades. Entretanto, é necessário que a função de atualização da posição e movimentação da sonda (ver item 2.3) seja chamada a cada 100ms.

Para realizar a inicialização das interrupções o Programa Principal fará, logo no início e nessa ordem, as seguintes operações:

- Inicializar o R6;
- Chamar a Rotina de Inicialização (ver item 3). Essa rotina é responsável por realizar todas as inicializações necessárias à operação da interrupção e deve ser programada nessa parte do trabalho.
- Habilitar as interrupções.

Nessa primeira parte do trabalho você deve se preocupar em implementar adequadamente a Rotina de Inicialização. Para realizar o teste de sua implementação será utilizado um Programa Principal que fará, exatamente, as tarefas indicadas acima.

5. Variáveis de Comunicação

Estão previstas seis variáveis que serão usadas na troca de informações entre os três componentes do programa. Essas variáveis são as seguintes:

- ON:** (16 bits) É a forma como se informa à sonda para ligar ou desligar o motor. Se essa variável valer zero, o motor será desligado. Caso contrário, a sonda ligará o motor.
- AR:** (16 bits) É a forma como se informa à sonda para acelerar ou reduzir. Se essa variável valer zero, a sonda reduzirá sua velocidade. Caso contrário, a sonda irá acelerar.
- TICK:** (16 bits) Informa que passou 1 (um) segundo. O tratador de interrupção deve escrever, a cada segundo, um valor diferente de zero nessa variável.
- GPS:** (16 bits) Contém a informação do GPS da sonda (ver item 2.3). Com essa informação pode-se determinar a distância entre a mesma e seu local de destino. Essa variável informa, em “nanosegundos”, o tempo de propagação do sinal de rádio enviado entre a sonda e o local de destino. A partir desse tempo e da velocidade das ondas de rádio no espaço, pode-se determinar a distância entre eles, aplicando-se a expressão $d=v.t$. No caso da sonda, para transformar essa medida de tempo em distância basta multiplicá-la por 300. O resultado dessa multiplicação deve ter 32 bits e ser colocado na variável “**DISTANCIA**”.
- DISTANCIA:** (32 bits) Essa variável representa a distância entre a sonda e o seu destino. Seu valor deverá ser calculado pelo seu tratador de interrupção. Por uma questão de precisão das informações, o resultado desse cálculo estará em milímetros, conforme descrição da variável “**GPS**”.

Além dessas variáveis existem outras duas: CARGA e VELOCIDADE, que serão fornecidas pela sonda e serão usadas pelo Programa Principal. Elas serão descritas na especificação da segunda parte do trabalho.

A forma como essas variáveis se relacionam e o módulo onde cada uma se encontra está representado na Figura 1. As setas no diagrama representam quem escreve nas variáveis e quem lê as variáveis. Assim, a **sonda** lê as variáveis “ON” e “AR” e escreve nas variáveis “GPS”, “CARGA” e “VELOCIDADE”. O **módulo de interrupção** lê a variável “GPS” e escreve nas variáveis “DISTANCIA” e “TICK”. Finalmente, o **Programa Principal** escreve nas variáveis “ON” e “AR” e lê as variáveis “DISTANCIA”, “TICK”, “CARGA” e “VELOCIDADE”.

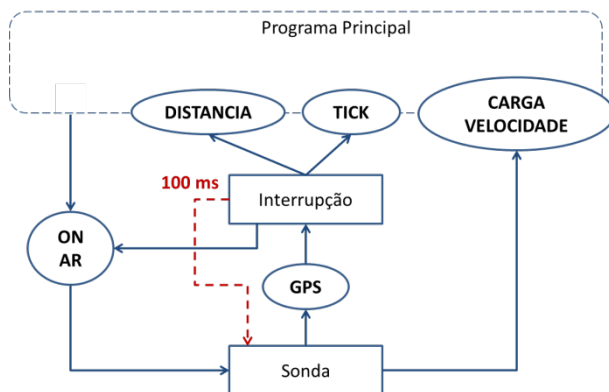


Figura 1 – Diagrama de representação da interação entre variáveis

6. Correção e Entregáveis

A correção dessa primeira parte será feita através de um programa principal especialmente construído para efetuar a correção automática da implementação do aluno. Portanto, a leitura e escrita das variáveis de comunicação devem ser realizadas **exatamente** conforme especificado.

A nota final do trabalho será proporcional aos casos de teste para os quais as implementações forem bem sucedidas. Essa primeira parte do trabalho terá **peso 4 sobre 10** enquanto que a segunda parte terá **peso 6 sobre 10**.

Cada parte do trabalho deverá ser entregue na entrada adequada do Moodle da turma. Deve ser entregue um arquivo fonte (arquivo .CED) com a solução correspondente, escrito em *assembly* do CESAR16i. Além disso, esse programa fonte deverá conter comentários descritivos da implementação. Sugere-se usar comandos da linguagem "C".

Cada parte do trabalho deverá ser entregue até a data prevista. Não serão aceitos trabalhos entregues além do prazo estabelecido. Trabalhos não entregues até a data prevista receberão nota zero.

7. Observações

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplinar Discente e a tomada das medidas cabíveis para essa situação (**tanto o trabalho original quanto os copiados receberão nota zero**).

O professor da disciplina reserva-se o direito, caso necessário, de solicitar uma demonstração do programa, onde o aluno será arguido sobre o trabalho como um todo. Nesse caso, a nota final do trabalho levará em consideração o resultado da demonstração.