

Sistemas Embarcados

Apresentação



Prof. Dr. Fábio Rodrigues de la Rocha

Sumário

- Sobre o que trata a disciplina ?

Sumário

- Sobre o que trata a disciplina ?
- Página da disciplina
<http://fabiodelarocha.paginas.ufsc.br/>

Sumário

- Sobre o que trata a disciplina ?
- Página da disciplina
<http://fabiodelarocha.paginas.ufsc.br/>
- Aulas, provas, trabalhos

Sumário

- Sobre o que trata a disciplina ?
- Página da disciplina
<http://fabiodelarocha.paginas.ufsc.br/>
- Aulas, provas, trabalhos
- Aulas de laboratório e teoria

Sumário

- Sobre o que trata a disciplina ?
- Página da disciplina
<http://fabiodelarocha.paginas.ufsc.br/>
- Aulas, provas, trabalhos
- Aulas de laboratório e teoria
- Programação em C

Sumário

- Sobre o que trata a disciplina ?
- Página da disciplina
<http://fabiodelarocha.paginas.ufsc.br/>
- Aulas, provas, trabalhos
- Aulas de laboratório e teoria
- Programação em C
- Plano de ensino

Sumário

- Sobre o que trata a disciplina ?
- Página da disciplina
<http://fabiodelarocha.paginas.ufsc.br/>
- Aulas, provas, trabalhos
- Aulas de laboratório e teoria
- Programação em C
- Plano de ensino
- Frequência

Introdução

Sistema Embarcado/Embutido

É um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla. Diferente de computadores de propósito geral, como o computador pessoal, um sistema embarcado realiza um conjunto de tarefas predefinidas, geralmente com requisitos específicos.

Introdução - Exemplos

- Vídeo-game;
- Forno de micro-ondas;
- Máquina de lavar louça;
- Robôs;
- Braços mecânicos;
- Câmaras fotográficas;
- Telefones celulares;
- Agendas eletrônicas;
- GPS;

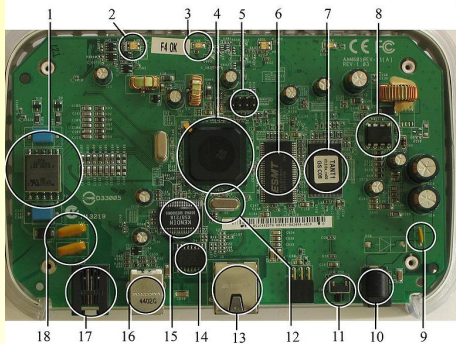
Introdução - Outros exemplos

- Players de MP3;
- CD Player;
- Modem;
- Impressoras;
- Máquinas de corte a laser;
- Balanças de precisão;
- PDAs;
- CLPs;
- Caixas eletrônicos de bancos;

Introdução - Outros exemplos



Introdução - Exemplo de um sistema Embarcado



Introdução - Requisitos

Funcionais

- Receber/transmitir dados ADSL
- Receber/transmitir dados PC
- possuir 4 conectores Ethernet
- suportar cabo ethernet comum e crossover
- Interface de configuração WEB

Introdução - Requisitos

Não-Funcionais

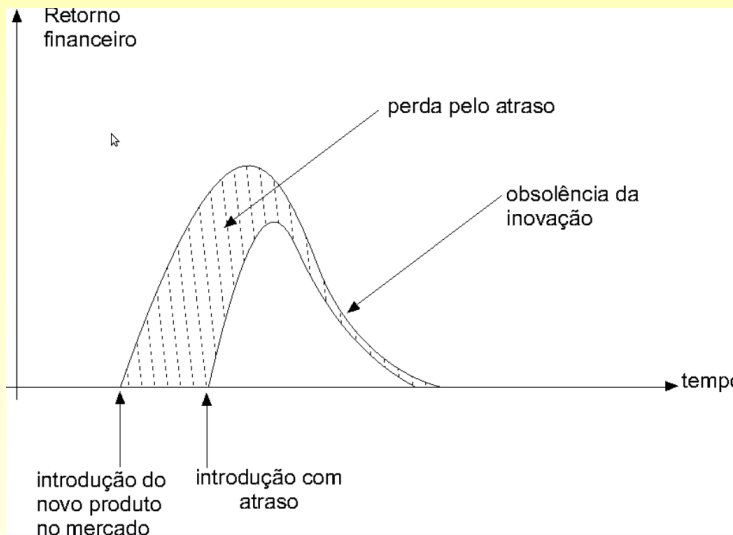
- Ocupar pouco espaço
- Conector de energia padrão, bivolt
- Gastar pouca energia
- Baixo custo
- Responder rapidamente ao usuário
- Ter interface gráfica amigável
- Possuir um design bonito

Introdução - Tendências

Tendências para o futuro

- Uma estimativa atual diz uma residência típica americana possui 35 produtos baseados em sistemas dedicados. Esse número deverá crescer para 250, nos próximos anos.
- Automóveis;
- Medicina;
- Cuidado de Idosos/Crianças.

Introdução - Mercado



Introdução - Erros em sistemas embarcados

falhas de projetos

- O que aconteceu em Marte ?
- Automóveis;
- Medicina;

Redução de consumo de recursos naturais:

- sistemas de aquecimento modernos
- redução de consumo de energia em motores elétricos (dado: 50% de toda eletricidade produzida no planeta é consumida por motores)

Introdução

- Microcontroladores qual da diferença entre microcontrolador para microprocessador (CPU) ? O que são ? características de microcontroladores ?

Introdução

- Microcontroladores qual a diferença entre microcontrolador para microprocessador (CPU) ? O que são ? características de microcontroladores ?

microcontrolador

Um microcontrolador é um microprocessador com algumas unidades funcionais extra:

- Memória RAM;
- Memória ROM;
- contadores, timers, PWMs, controlador de interrupção;
- barramentos (Serial, I2C, USB, SPI, CAN);
- Watchdog timer;

microcontrolador

- Velocidade de processamento: de 8 a 20 MHz (usual)
- Capacidade de memória:
 - memória interna de programa : até 64K (usual)
 - memória interna de dados : até 1K (usual)
 - memória externa de dados : até alguns MBytes
- Conversor AD e DA;
- Pinos de E/S programáveis;

microcontrolador

- Modo de baixo consumo: - 1uA (usual)
- Sistemas de segurança (não usual) - criptografia
- Ferramentas de software incorporadas (não usual):
 - monitor residente
 - interpretador BASIC residente

Introdução - Programação

Linguagem Assembly:

- semelhante à dos microprocessadores tradicionais;
- modos de endereçamento variados

Linguagem BASIC:

- fácil de aprender
- interpretada, lenta
- simplificação no uso de variáveis, gastos com memória

Linguagem C:

- disponível em 1985 para microcontroladores
- código relativamente compacto e rápido
- sponteiros, bitfields, modular, tipologia

Linguagem C++, Java:

Introdução - Desenvolvimento

- crosscompilers
- compiladores livres/comerciais;
- kits didáticos;
- ambientes integrados de desenvolvimento.

Introdução - Arquitetura

- von Neumann:
 - execução seqüencial de instruções
 - barramento único para instrução e dados;
- Harvard:
 - maior paralelismo
 - barramentos separados para instrução e dados

Introdução - Arquitetura

Processadores digitais de sinais, composição (usual):

- multiplicador por hardware;
- múltiplas unidades de execução
- pipeline sofisticado
- arquitetura Harvard

Características (gerais):

- arquitetura dedicada ao processamento de sinais
- altas taxas de processamento (execução em um ciclo)

Introdução - Fabricantes

Microcontroladores:

- Intel, Motorola (Freescale), MicroChip, National, Texas, Zilog, Sharp, Analog Devices, Atmel, outras

DSPs:

- Texas, Motorola, Analog Devices, microchip

Introdução - Composição

- uma CPU, memória RAM, algum tipo de ROM;
- porta serial, portas paralelas, contadores e temporizadores;
- e um controlador de interrupções.
- Características (gerais):
- amplo conjunto de instruções para manipular bits
- acesso diretos aos pinos de e/s
- rápido e eficiente sistema de atendimento de interrupções

Introdução

Um computador pessoal também é composto por hardware e software, mas é projetado para realizar muitas tarefas diferentes, tais como processador de textos, servidor de arquivos, executar sistemas de análise numérica, etc. A empresa que fabrica um computador, não sabe qual será a aplicação que o usuário irá utilizar o computador.

Introdução

Tipicamente um sistema embutido é responsável pela execução de um laço de controle, uma máquina de estados ou um algoritmo de tratamento de sinais.

Introdução - Diferenças

Um sistema embutido é um projeto multidisciplinar, pois reúne pessoas de várias áreas tais como:

- Designer artístico
- Engenheiro elétrico
- Engenheiro mecânico
- Analista/Programador

Introdução - Diferenças

Sistemas embutidos possuem importantes diferenças em relação a computadores de propósito geral

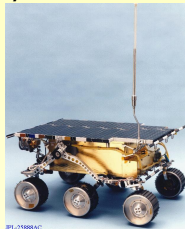
Interface com o usuário pode ser simples como LEDs piscando ou complexas como sistemas de visão de robôs

Introdução - Diferenças

- Para melhorar o desempenho podem ser utilizados em conjunto com hardware analógico, com FPGAs e ASICs;
- Software possui uma função bem determinada e específica para uma determinada aplicação
- Podem existir restrições quanto ao consumo de energia, formato, tamanho, peso e custo.

Introdução - Diferenças

Outras restrições também podem estar presentes como resistir a altas/baixas temperaturas, ambientes corrosivos, campos magnéticos, trepidação, tolerância a falhas,



atualizações, logística, etc.
Exemplo: Sonda Pathfinder

Introdução - Quanto a construção

Um computador pessoal também pode ser utilizado para construir um sistema embutido. Neste caso, seu objetivo não é ser um computador de propósito geral mas realizar tarefas específicas. Utilizar um computador de propósito geral para construir um sistema embutido possui vários benefícios, tais como: todo o hardware já está desenvolvido (teclado, display, porta serial, porta paralela) existem compiladores, bibliotecas de software e existem dispositivos de armazenamento RAM e discos magnéticos. Ex: Controlar um robô

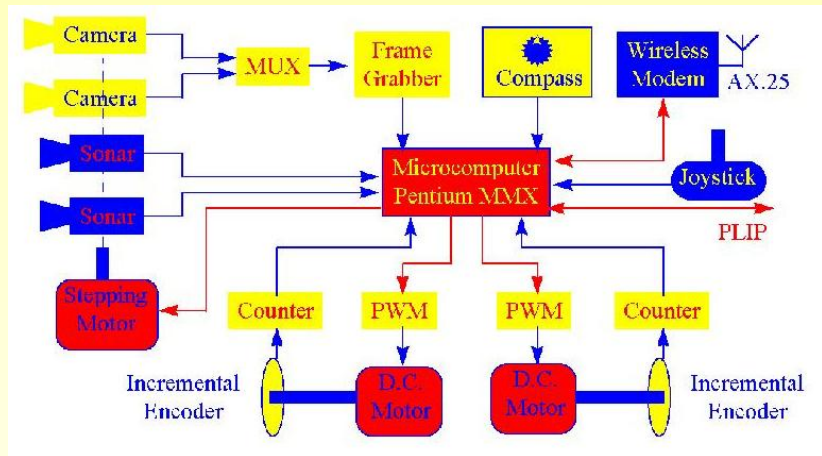
Introdução - Quanto a construção



Introdução - Quanto a construção



Introdução - Quanto a construção



Introdução - Restrições

Geralmente um sistema embutido é constituído de um dispositivo simples. Onde simples significa o hardware minimamente necessário para executar determinada aplicação. Seria difícil tentar justificar a utilização de microprocessador P4 para construir um forno de microondas ou uma lavadora de louça e mais difícil ainda conseguir vender o produto com preço competitivo.

Introdução - Restrições

Desta forma, os sistemas embutidos possuem limitações de processador (taxa de execução de instruções), de memória (tamanho da memória RAM e ROM/FLASH).

Introdução - Restrições

Para lidar com restrição de CPU devem ser utilizados algoritmos cuja complexidade computacional seja baixa. Para lidar com restrições de memória deve-se somente inserir no sistema embutido o código que será utilizado.

Introdução - Restrições

No passado os dispositivos eram mais simples e programados com poucas centenas de linhas de código em assembly. Atualmente, esse quadro não se aplica, hoje os dispositivos possuem vários processadores e executam dezenas de milhares de linhas de código em linguagem de alto nível.

Ferramentas

Para lidar com essa complexidade e para continuar produzindo dispositivos num mundo cada vez mais competitivo, foram criadas técnicas e ferramentas para facilitar a criação desses dispositivos.

Ferramentas

Para construção de hardware: placas de desenvolvimento conectadas ao computador com emuladores de hardware, linguagens de descrição de hardware (VHDL), simuladores, etc.

Para construção de software: camadas de abstração de software (micro-kernel, SA) e ferramentas de modelagem e geração de código personalizado para determinado tipo de aplicação.

Muitos dispositivos embutidos são utilizados em aplicações de tempo real. Nestas aplicações o sistema embutido além de fornecer uma resposta correta, deve realiza-la dentro de um determinado período de tempo chamado 'Deadline'.

Tempo Real - previsibilidade

No sistemas de tempo real, a grande preocupação é o tempo de resposta conhecido para suas ações.

Tempo Real - previsibilidade

O software programado nestes sistemas precisa ter um tempo conhecido para todas as suas funções e as camadas de software que existem abaixo deste também precisam ter previsibilidade conhecida.

Concorrência e bloqueios

O escalonador (scheduler) escolhe qual thread deve ganhar direito ao processador. Num sistema multithread várias threads poderão estar executando o mesmo trecho de código “simultaneamente”.

Concorrência e bloqueios

Para manter a consistência de estruturas de dados é necessário garantir que somente uma thread pode estar ativa dentro de uma determinada seção crítica de código.

Inversão de prioridade

Em muitos casos, sistemas de tempo real utilizam escalonadores com política de prioridade. O escalonador pode interromper a execução de tarefas (threads) de prioridade mais baixa para em seu lugar colocar uma tarefa de prioridade mais baixa.

Inversão de prioridade

Quando uma tarefa de baixa prioridade está dentro de uma seção crítica, mesmo que o escalonador escolha substituí-la por outra tarefa mais prioritária esta não poderá acessar a seção crítica. Neste caso temos uma tarefa de baixa prioridade que impede a execução de uma tarefa de alta prioridade. Esta situação é chamada de inversão de prioridade.