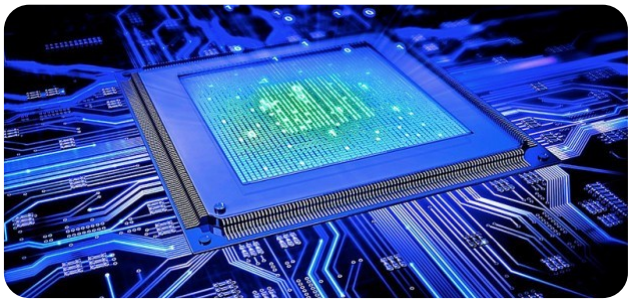
Introdução aos Sistemas Embebidos



Conteúdo a abordar

- O que são sistemas de computação embebidos
- Desafios no design de sistemas de computação embebidos
- Metodologias de design/conceção

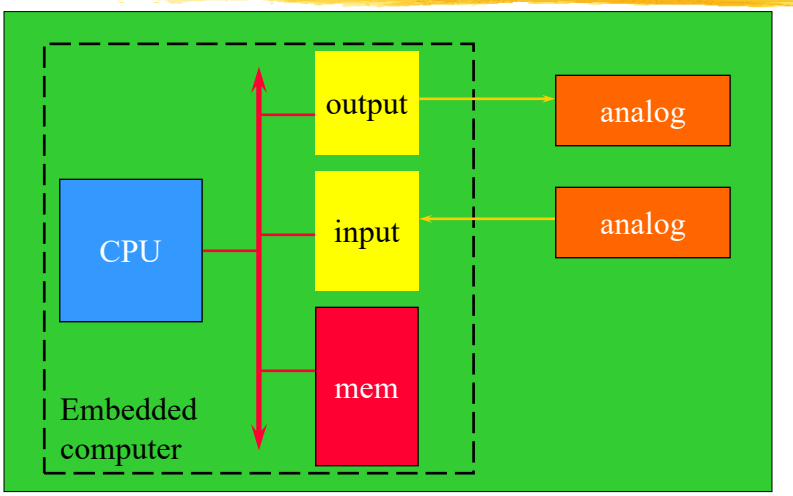
Definição

- Sistema de computação embebido: qualquer dispositivo que inclui um computador programável mas não é um computador de propósito geral.
- Não necessita de todas as caraterísticas de propósito geral (ex: interrupções, interfaces, etc.)
- É uma combinação de hardware e software personalizados que executam uma tarefa específica.
- É uma subcategoria da Internet das Coisas (IoT), dispositivos que podem ou não estar conectados à Internet.

3

3

Sistema de Computador Embebido



4

4

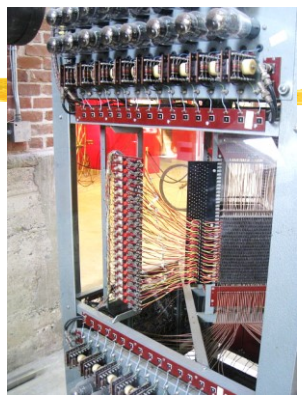
Exemplos

- Telemóveis
- Impressoras
- Automóveis: motor, travões, airbag, etc.
- Aviões: motor, controlo de voo, navegação/comando
- Televisão digital
- Eletrodomésticos

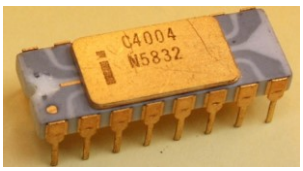
5

5

Histórico (1)



- Final dos anos 40: MIT desenvolve o computador ***Whirlwind*** para aplicações de tempo real
  - Originalmente desenvolvido para controlar um simulador de voo
- O primeiro microprocessador surgiu nos anos 70 - Intel 4004 (4bit 740 kHz)

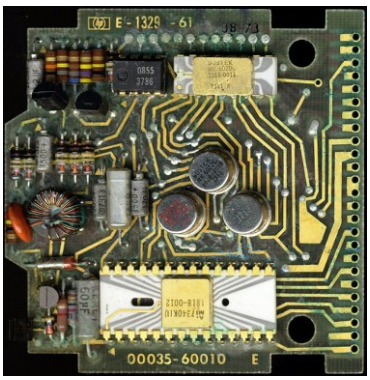


6

6

Histórico (2)

- Em 1972 a calculadora HP-35 utilizava vários ICs (circuito integrados) para implementar um microprocessador.



7

7

Histórico (3)

- Nos anos 70 também os automóveis começam a utilizar controladores do motor baseados em microprocessadores
  - Controlo da mistura combustível/ar, sincronismo do motor, etc.
  - Múltiplos modos de operação: aquecimento, cruzeiro, montanha, etc.
  - Proporciona emissões reduzidas, maior eficiência de combustível, etc.

8

8

Tipos de Microprocessadores

- Microcontrolador: inclui periféricos de I/O e memória integrada
- Processador de sinal digital (DSP): µprocessador otimizado para o processamento de sinais digitais (áudio/vídeo, etc,)
- Dimensão típica das *word* nos sistemas embebidos: 8-bit, 16-bit, 32-bit

9

9

Exemplos de aplicações

- Controlo simples: painel frontal do forno micro-ondas, etc.
- Máquina fotográfica - *Canon EOS3* tem 3 µprocessadores
  - *CPU 32-bit RISC* executam o autofócus e controlo ocular
- TV Digital: *CPUs* programáveis + *hardware* lógico para descodificação de vídeo/áudio, menus, etc.

10

10

Sistemas Embebidos em Automóveis

- Os automóveis atuais podem integrar mais de 100 microprocessadores:
  - Microcontroladores de 4-bits verificam os cintos de segurança
  - Microcontroladores operam os dispositivos de instrumentação
  - Um microprocessador de 16/32-bit controla o motor

11

11

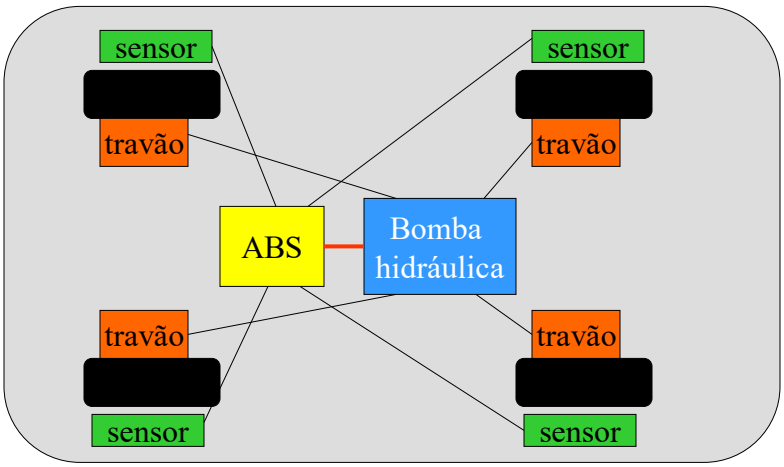
BMW 850i - Sistema de Travagem e Controlo de Estabilidade

- ABS: sistema hidráulico de travagem intermitente para reduzir a derrapagem
- Controlo automático de estabilidade e tração (ASC+T): controlo do motor e travões para aumentar a estabilidade
- ABS e ASC+T comunicam
  - ABS foi introduzido primeiro, necessidade de interface com módulos de ABS existentes

12

12

BMW 850i (2)



13

Caraterísticas dos Sistemas Embebidos

- Funcionalidade sofisticada
- Operação em tempo real
- Baixo custo de fabrico
- Baixo consumo
- Projetados para deadlines reduzidos por equipas pequenas

14

Operação em Tempo Real

- Devem finalizar operações com <u>restrições temporais</u>
  - *Hard real time:* o não cumprimento das restrições causa a falha da ação
  - *Soft real time:* o não cumprimento das restrições causa redução da performance

15

15

Requisitos Não-Funcionais

- Muitos SE são itens de mercado de massa que devem ter baixo custo de fabrico.
  - Memória limitada, baixa freq. processador…
- O consumo de energia é crítico em dispositivos alimentados por bateria.
  - O consumo excessivo de energia aumenta o custo do sistema (melhores sistemas de alimentação).

16

16

Equipas de projeto

- Normalmente desenvolvidos por equipas de poucos elementos
- Habitualmente devem cumprir prazos apertados
  - Janela de mercado de 6 meses é comum
  - Pouco espaço para falhas

17

17

Porque utilizar Microprocessadores?

- Alternativas: *field-programmable gate arrays* (*FPGAs*), lógica customizada, etc.
- Microprocessadores são muito eficientes: podem utilizar a mesma lógica para realizar diferentes funções
- Microprocessadores simplificam o desenvolvimento de famílias de produtos

18

18

O Paradoxo da Performance

- Microprocessadores utilizam muito mais lógica para implementar uma função do que a lógica customizada
- Contudo os microprocessadores são tipicamente mais rápidos:
  - Fortemente *pipelined*;
  - Concebidos por grandes equipas de desenvolvimento;
  - Tecnologia *VLSI* agressiva.

VLSI = Very-Large-Scale Integration

19

Energia

- A lógica customizada requer menos energia, mas os *CPUs* possuem vantagens:
  - Microprocessadores modernos possuem caraterísticas que permitem controlar o consumo de energia
  - Técnicas de design de software podem ajudar a reduzir o consumo de energia
- Sistemas heterogéneos: lógica customizada para funções bem definidas, *CPUs* + software para tudo o resto.

20

Plataformas

- Plataforma de computação embebida:
  - arquitetura de hardware + software associado
- Muitas plataformas integram vários microprocessadores

Exemplos:

- Multiprocessadores em telemóveis
- Redes de processadores em automóveis

21

O Que Significa "Performance"?

- Na computação de propósito geral, performance significa *throughput* médio, que pode não ser bem definido.
- Em sistemas tempo real, performance significa cumprir as restrições
  - Falhar o *deadline* pode resultar em consequências graves
  - Término após o *deadline* pode ser igual a não executar!

22

Caraterização da Performance

- Necessitamos de analisar o sistema a vários níveis de abstração para compreender a performance:
  - CPU
  - Plataforma
  - Programa
  - Tarefas
  - Multiprocessadores

Desafios no desenvolvimento de Sistemas Embebidos (1)

- Qual a quantidade de hardware necessária?
  - Qual a frequência do CPU? Memória?
- Como cumprir as restrições temporais?
  - Hardware rápido ou software inteligente?
- Como minimizar o consumo de energia?
  - Desativar lógica desnecessária? Reduzir acessos à memória?

Desafios no desenvolvimento de Sistemas Embebidos (2)

- Funciona realmente?
  - A especificação está correta?
  - A implementação está de acordo com a especificação?
  - Como testar para características de tempo real?
  - Como testar com dados reais?
- Qual o princípio de ação no sistema?
  - Monitorização, controlo?
  - Qual é a plataforma de desenvolvimento?

25

Metodologias de conceção

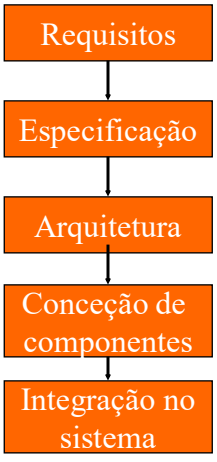
- Procedimento para projetar um sistema
- Compreender a metodologia de desenvolvimento ajuda a garantir que nada é suprido
- Compiladores, ferramentas de engenharia de software, ferramentas de CAD, etc., podem ser utilizadas para:
  - Automatizar as etapas da metodologia;
  - acompanhar a própria metodologia.

26

Objetivos de conceção

- Performance
  - Velocidade, restrições temporais
- Funcionalidade e interface com o utilizador
- Custo de fabrico
- Consumo de energia
- Outros requisitos (dimensão física, materiais, etc.)

Níveis de Abstração



Top-down vs. *bottom-up*

- Conceção *top-down*:
  - Começa com a descrição mais abstrata;
  - Alcança a descrição mais detalhada.
- Conceção *bottom-up* :
  - Inicia com componentes pequenos até atingir um sistema de grandes dimensões.
- Tipicamente são utilizadas ambas as técnicas!

Refinar Passo-a-Passo

- A cada nível de abstração deve-se:
  - Analisar o design para determinar características do estado corrente de design;
  - Refinar o design para acrescentar detalhe.

Requisitos

- Descrição em linguagem corrente das necessidades e expetativas do utilizador.
- Podem ser obtidos de várias formas:
  - Contacto direto com os clientes;
  - Contacto direto com os representantes de marketing;
  - Disponibilizar protótipos para comentário dos utilizadores.

31

31

Requisitos Funcionais vs. Não-Funcionais

- Requisitos funcionais:
  - indica o que o sistema deve fazer, em termos de tarefas e serviços - *output* como uma função de *input*
- Requisitos não-funcionais:
  - Tempo necessário para processar o *output*;
  - Tamanho, peso, etc.;
  - Consumo de energia;
  - Fiabilidade, etc.

32

32

Formulário de Requisitos

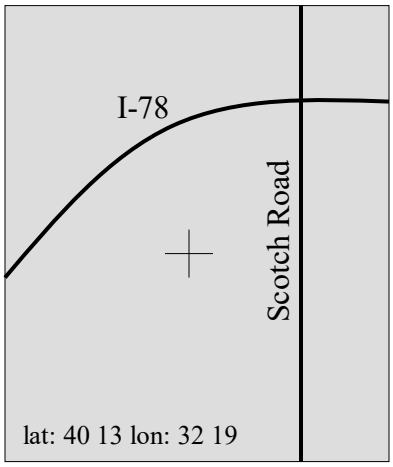
nome
propósito
inputs
outputs
funções
performance
custo fabrico
consumo energia
dimensão/peso

33

33

Exemplo: requisitos de um mapa de GPS (1)

- As coordenadas GPS identificam a posição no mapa, definem o traçado a partir de uma base de dados local.



34

34

Exemplo: requisitos de um mapa de GPS (2)

- Funcionalidade: Para uso automóvel. Apresenta principais estradas e marcos.
- Interface com o utilizador: pelo menos um ecrã de 400 x 600 pixéis. Três botões no máximo. Menu pop-up.
- Performance: O mapa deve varrer-se lentamente. Menos de 1 segundo para ligar. Estabilizar no GPS em 15 segundos.
- Custo: 100€ preço venda mercado = aprox. 30€ custo de venda fábrica

35

35

Exemplo: requisitos de um mapa de GPS (3)

- Dimensão/Peso: Deve caber numa mão
- Consumo de energia: Autonomia de 8 horas com 4 pilhas AAA.

36

36

Exemplo: requisitos de um mapa de GPS (4)- Formulário

Nome	GPS mapa
Propósito	Mapa movimento para condução
Inputs	Botão de energia, 2 botões de controlo
Outputs	LCD 400 X 600
Funções	Recetor de 5 sat. GPS; três resoluções; apresenta lat/lon actual
Performance	Actualizar o ecrã a cada 0.25 s de movimento
Custo de fabrico	100€
Consumo energia	100 mW
Dimensão/Peso	< 5 X 12 cm, 400 g

37

37

Especificação

- Descrição mais precisa do sistema:
  - Não deve implicar uma arquitetura particular;
  - permite *input* ao processo de desenvolvimento da arquitetura.
- Pode incluir elementos funcionais e não funcionais
- Pode ser executável ou em forma matemática para prova

38

38

Especificação do GPS

- Deve incluir:
  - O que é recebido pelo GPS;
  - Dados do mapa;
  - Interface com o utilizador;
  - Operações necessárias para satisfazer os pedidos do utilizador;
  - Operações de *background* necessárias para manter o sistema em execução.

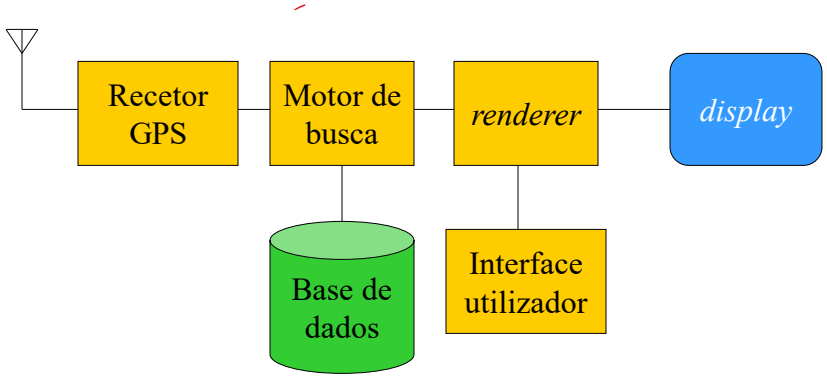
39

Desenvolvimento da Arquitectura

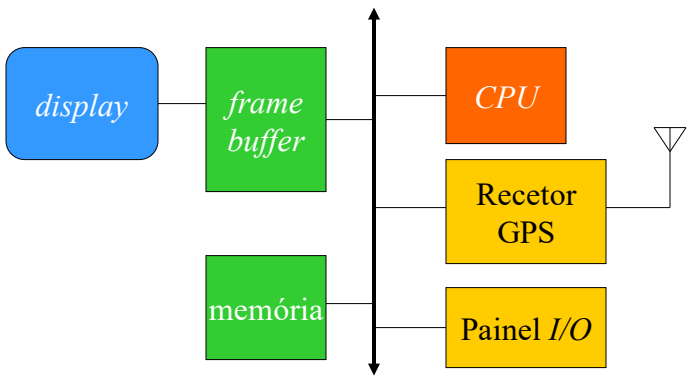
- Quais os componentes fundamentais para satisfazer a especificação?
- Componentes de hardware:
  - *CPUs*, periféricos, etc.
- Componentes de software:
  - Principais programas e suas operações
- Deve considerar as especificações funcionais e não-funcionais

40

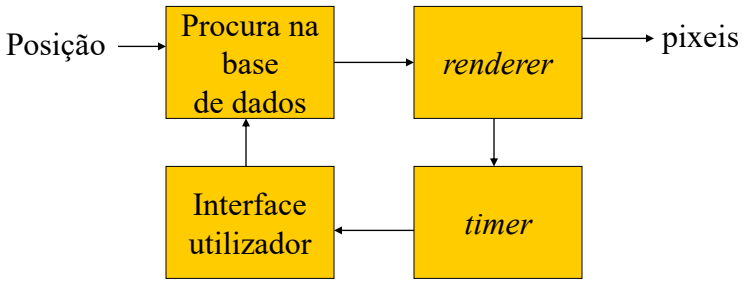
Diagrama de Blocos - GPS



Arquitetura de hardware - GPS



Arquitetura de software - GPS



Conceção de componentes de hardware e software

- Antes da codificação é necessário arquitetar o sistema
  - Alguns componentes já existem,
  - outros podem ser alterados de projetos existentes,
  - outros têm de ser totalmente desenvolvidos

Integração do Sistema

- Juntar os vários componentes do sistema
  - Surgem vários bugs nesta fase
- Deve possuir-se um plano para detetar bugs rapidamente
- Deve testar-se a máxima funcionalidade tão cedo quanto possível

45

45

Resumo

- Os computadores embebidos fazem parte do nosso dia-a-dia
  - Muitos sistemas possuem hardware e software embebido complexo
- Os sistemas embebidos apresentam vários desafios de desenvolvimento: tempo de conceção, restrições, energia, etc.
- As metodologias de desenvolvimento ajudam a gerir o processo de conceção

46

46