

# Especificação, Modelação e Projecto de Sistemas Embutidos

## Apresentação da disciplina

Paulo Pedreiras, Luís Almeida

{pbrp,lda}@ua.pt



Departamento de Electrónica, Telecomunicações e Informática  
Universidade de Aveiro

# Objectivo da Disciplina

- (Re)Conhecer a **especificidade** dos sistemas embutidos (SE) e **identificar** as diversas vertentes das restrições que lhes aparecem normalmente associadas;
- Estudar as **arquitecturas** e **paradigmas** de software tipicamente usadas para o desenvolvimento de SE
- Conhecer os **recursos** de **hardware e software** usados habitualmente no desenvolvimento de SE e avaliar a sua adequação a situações concretas;
- Compreender o **ciclo de projecto** de SE;

# Objectivo da Disciplina

- Conhecer alguns dos **paradigmas** de **especificação e modelação** de sistemas embutidos mais representativos;
- Capacidade de **utilização** dos **conhecimentos adquiridos** nas diversas fases do projecto de sistemas embutidos, envolvendo nomeadamente a especificação, a modelação, a simulação e a síntese.

# Começando pelo principio ...

- Algumas questões iniciais:
  - Afinal **o que são** “sistemas embutidos”?
  - **Para que servem** os sistemas embutidos?
  - Em que é que os sistemas embutidos **diferem** dos restantes sistemas computacionais?
  - ... e finalmente, em que medida é que o que vou aprender nesta disciplina pode afectar a **minha felicidade**?

# Definição de Sistema Embutido

- Há diversas definições para “Sistema Embutido”, e.g.:
  - **Sistema computacional** composto por hardware, software e (eventualmente) componentes mecânicos, projectado para **desempenhar apenas uma função específica**, fazendo frequentemente parte de um produto ou sistema mais vasto e complexo.
    - (adaptado de `http://www.netrino.com/Embedded-Systems/Glossary-E```)
  - Sistema computacional parte de dispositivos em que a **existência** de um **computador não é óbvia**.
  - Num tom mais jocoso pode adaptar-se esta última definição dizendo que “estamos na presença de um SE quando há uma falha num equipamento causada por um **problema** num **computador** do qual se **desconhecia** em absoluto a **existência**”.

# Áreas de aplicação

- Áreas de aplicação dos SE
  - Na actualidade os SE estão presentes em **quase todos os aspectos do nosso quotidiano**. Sem ser exaustivo podem referir-se a título de exemplo:
    - **Electrónica de consumo:**
      - PDAs, MP3/4, consolas de vídeo-jogos, câmeras digitais (fotográficas e de filmar), impressoras, máquinas de lavar, torradeiras, frigoríficos, fornos microondas, aparelhos de ar condicionado, ...
    - **Telecomunicações:**
      - equipamentos de rede (e.g. *routers*, *switches*, etc.), terminais (e.g. telemóveis), centrais de comutação, ...
    - **Sistemas de transporte:**
      - veículos automóveis , gerindo sistemas de controlo do motor (e.g. injeção) e segurança (e.g. ABS, ESP), sistemas de gestão de tráfego (e.g. semáforos), ...
    - **Equipamentos médicos:**
      - sistemas de monitorização de sinais vitais, sistemas de imagiologia médica, controlo de equipamentos de tratamento (e.g. radioterapia), ...

# Áreas de aplicação

Algumas estatísticas interessantes sobre o uso de ES:

- Um **lar** americano **médio** tinha em 1999 **para cima de 40 microprocessadores** (50, se tiver PC) (\*) em equipamentos como:
  - Balança electrónica
  - Ferro de passar automático
  - **Escova de dentes electrónica (cerca de 3000 linhas de código!!!)**
  - Máquina de lavar roupa e louça
  - Forno microondas
  - Brinquedos, ...
- Em 1998 o número total de processadores vendidos foi de cerca de 4.8 biliões. Destes **apenas 2.5% se destinaram ao mercado de PCs (\*\*)**

(\*) Honey, I programmed the blanket, Katie Hefner, New York Times, 1999

(\*\*) Statistics from Dataquest

# Caracterização dos SE

Mas afinal o que distingue um SE de um sistema computacional vulgar?

- O **número de funções** que executam é **limitado** (dedicados)
- **Hardware é customizado**; apenas os componentes necessários para servir a aplicação são empregues;
- O software de aplicação é **adaptado especificamente** ao sistema
- Os SE têm **restrições severas** em termos de performance, memória, consumo de energia, ...
- Frequentemente são empregues no **controlo de sistemas físicos**, tendo de **reagir** a variações no sistema em intervalos de tempo curtos e determinísticos
- Podem possuir **requisitos de fiabilidade extremos** (e.g. é complicado efectuar manutenção a um SE presente num satélite)

**Estas restrições não estão habitualmente presentes nos sistemas computacionais vulgares!**



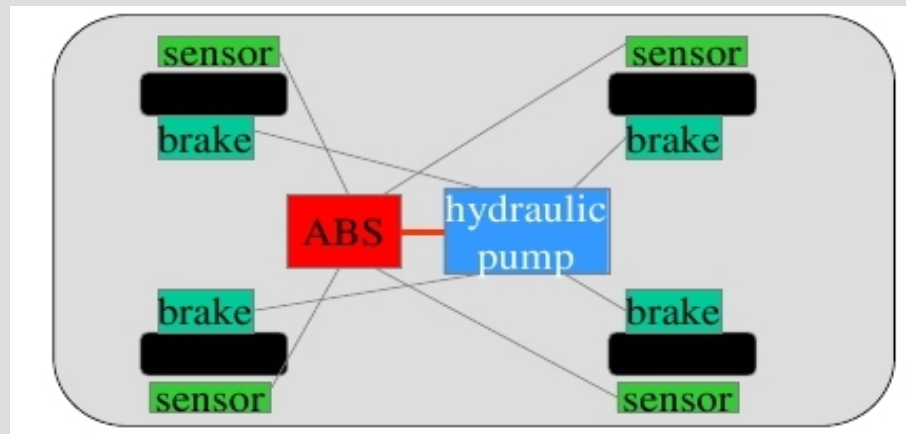
# Caracterização dos SE

Para além disso apresentam restrições como:

- **Quantidade de memória limitada**
  - Pode representar fatia significativa dos custos
- **Restrições em termos de tamanho e peso**
  - e.g. PDAs, MP3/4, telemóveis
- **Baixo consumo de energia**
  - alimentação por baterias, dificuldades de dissipação de calor
- **Desempenho tem de ser suficiente**
  - tempo de resposta frequentemente limitado, e.g. ABS
- **Baixo custo**
  - e.g. um carro médio emprega acima de 50 ECUs
- **Ambientes agressivos**
  - vibração, choque, temperaturas externas, etc (e.g. automóveis)
- **Segurança crítica:** em diversas aplicações ... **não podem falhar!**

# Caracterização dos SE

- E são frequentemente **reactivos** ...

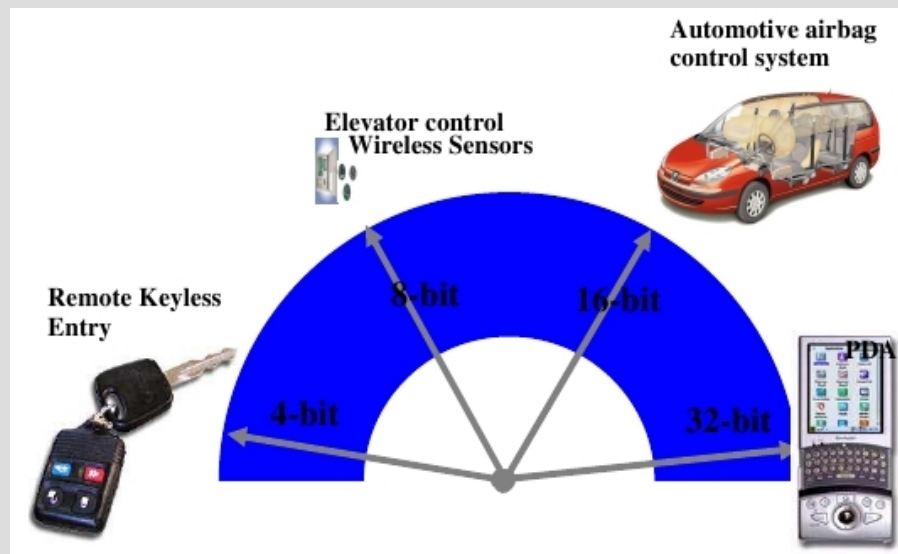


Fonte: "Computers as Components", W. Wolf

- Muitos ES são empregues em malhas de controlo, tendo de **reagir** “suficientemente depressa” às **variações do ambiente**
- Neste exemplo tem de observar a velocidade angular de cada roda e “aliviar” a pressão do travão quando estas rodam demasiado devagar
- **Sistema com requisitos de pontualidade, fiabilidade e segurança!**

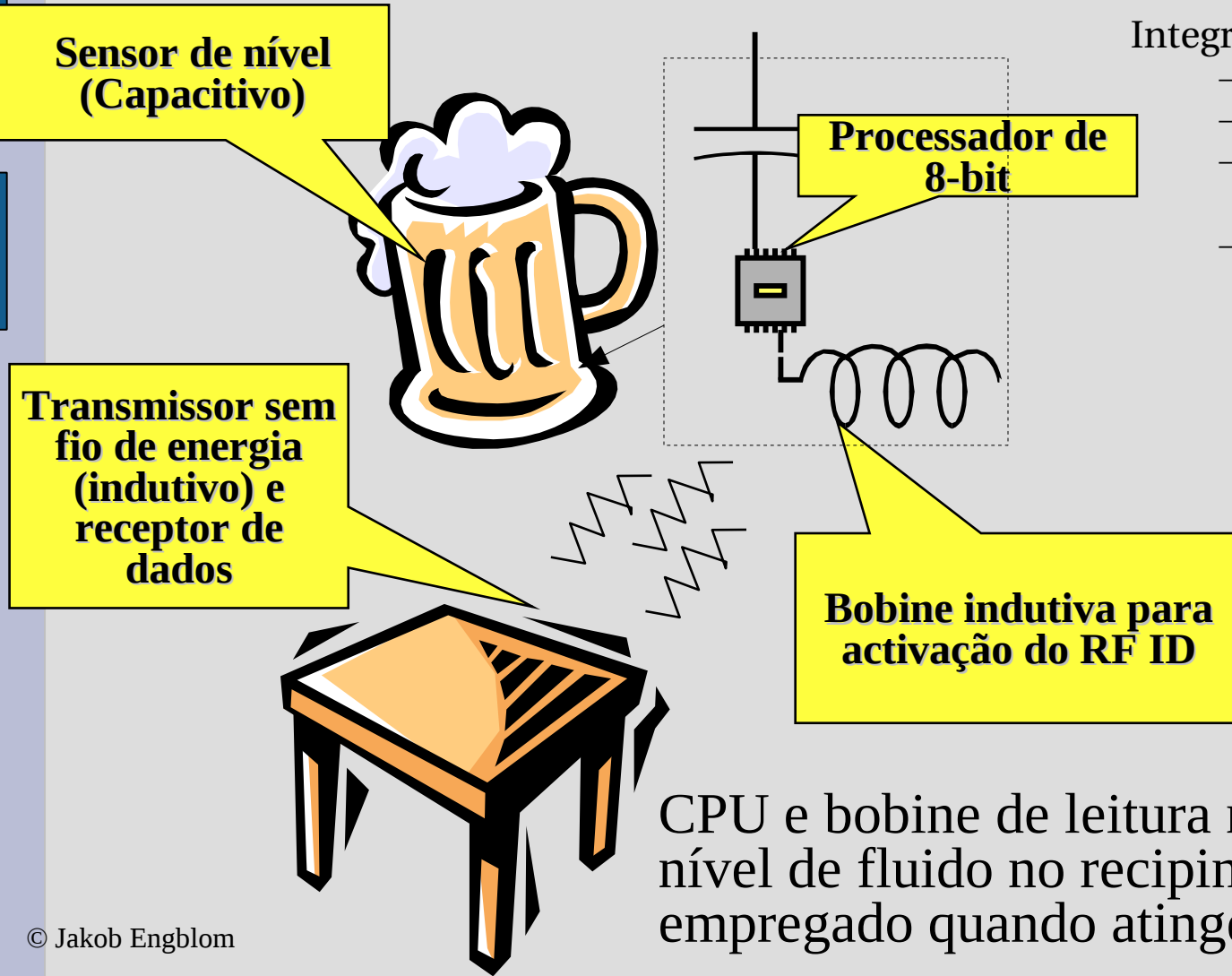
# Caracterização dos SE

- Sem esquecer que os ES são **heterogêneos** (aplicações, software e hardware) ...



- Uma lista (não exaustiva!!) de **arquitecturas comumente usadas** em ES: 65816, 65C02, 68HC08, 68HC11, 68k, 8051, ARM, AVR, AVR32, Blackfin, C167, Coldfire, COP8, eZ8, eZ80, FR-V, H8, HT48, M16C, M32C, MIPS, MSP430, PIC, PowerPC, R8C, SHARC, ST6, SuperH, TLCS-47, TLCS-870, TLCS-900, Tricore, V850, x86, XE8000, Z80, etc.

# Exemplo de um SE ...



Integra diversas tecnologias:

- Transmissão rádio
- Tecnologia de sensores
- Indutância magnética para energia
- CPUs (leitura de sensores, processamento de dados e comunicação)

CPU e bobine de leitura na mesa. Lê o nível de fluido no recipinete e chama o empregado quando atinge nível baixo

# Caracterização dos SE

E quais são as consequências de todas estes requisitos e restrições?

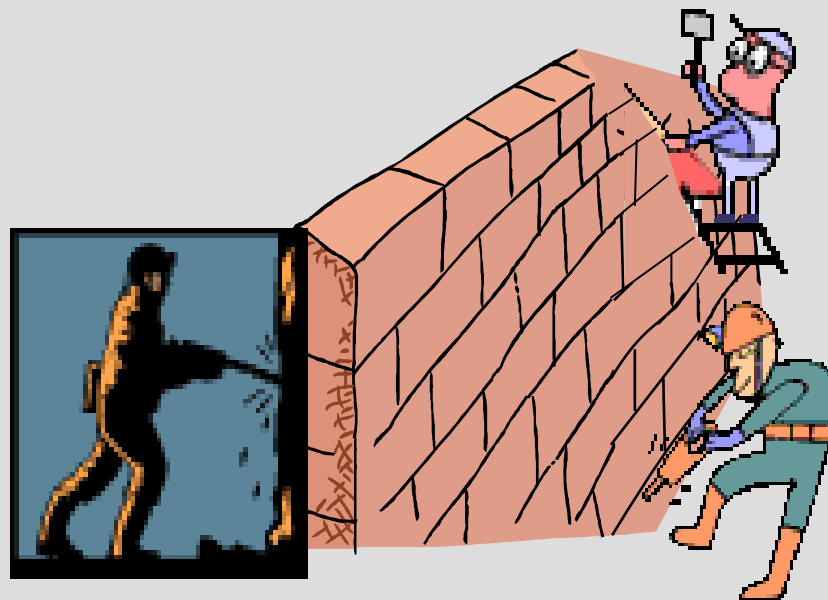
- Necessidade de **métodos e ferramentas específicas** para:
  - A **especificação** de ES
  - A **modelação** de ES
  - A **implementação** de ES
  - A **validação** de ES

# Enquadramento

E esta opção não é para “pessoal de ET” / pessoal de “ECT” ?

CS

EE



Área multi-disciplinar/transversal!  
Complementaridade entre ambas as áreas.

**É para ambos!**

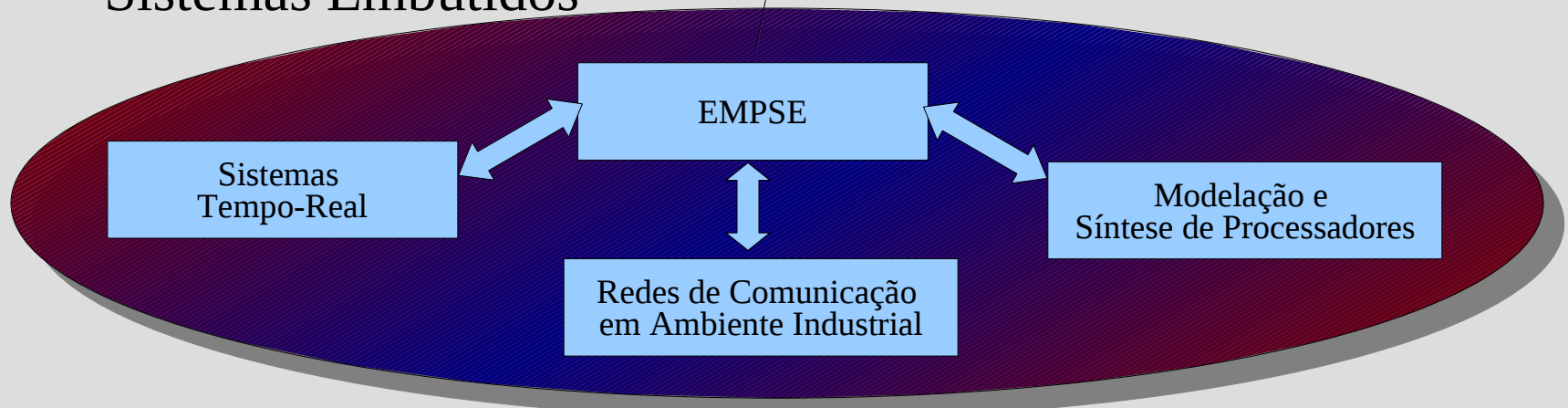
# Enquadramento

Enquadramento:

## Formação inicial:

- Arquitectura de Computadores
- Algoritmos/Programação
- Matemática
- Física
- Redes
- ...

Sistemas Embutidos



# Saídas profissionais

Ah! E afinal em que é que o que vou aprender nesta disciplina pode afectar a minha felicidade?

- Para além de ficar com medo (ou mesmo pânico) de andar de avião, de acidentes em centrais nucleares, indústrias químicas (entre outros medos ...), a área de ES proporciona **múltiplas saídas profissionais**:
  - Desenvolvimento de equipamentos de rede, “*appliances* domésticas”, ...
  - Sistemas de transporte e veiculares
  - Sistemas de controlo e supervisão em indústrias de processo e manufactura, ...
  - Desenvolvimento de maquinaria industrial
  - Indústria aeroespacial, indústria de defesa
  - ...



# Dados da disciplina

Passando agora a assuntos mais burocráticos ...

- Dados da disciplina:

- Nome: **Especificação, Modelação e Projecto de Sistemas Embutidos**
- Responsáveis: **Paulo Pedreiras, Luís Almeida**
- Contactos: **DETI Gab. 321, {pbrp,lda}@ua.pt**
- *Site*: **<http://www.ieeta.pt/lse/empse>**
- Cursos: **MIET/MIECT**
- Área/Sub-área Científica: **Informática / Ciência e Tecnologia da Programação**

# Objectivos de formação

- Conhecer a **especificidade dos SE** por oposição a sistemas computacionais genéricos e identificar as diversas vertentes das restrições que lhes aparecem normalmente associadas;
- Estudar as **arquitecturas e paradigmas** de software tipicamente usadas para o **desenvolvimento** de SEs;
- Conhecer os **recursos de hardware e software** usados habitualmente no desenvolvimento de SEs e saber avaliar a sua adequação a situações concretas;
- Compreender o **ciclo de projecto de SEs**;
- Conhecer alguns dos **paradigmas de modelação** de SEs mais representativos;

# Objectivos de formação(2)

- Compreensão do papel da **simulação** no projecto e desenvolvimento de SEs;
- Conhecer as **vantagens e limitações** das ferramentas utilizadas;
- Capacidade de **utilização** dos conhecimentos adquiridos nas diversas fases do **projecto** de sistemas embutidos, envolvendo nomeadamente a especificação, a modelação, a simulação e a síntese.

# Funcionamento

## Organização do trabalho:

- Haverá diversos tipos de aulas e trabalhos:
  - **Aulas Teóricas**
    - carácter essencialmente expositivo. A participação é todavia bem-vinda (e valorizada)
  - **Aulas “hands on”**
    - familiarização com algumas ferramentas de modelação, simulação e síntese
  - **Aulas Práticas**
    - Desenvolvimento de mini-projectos em grupo (complementado com algum trabalho fora das aulas)
  - **Trabalho de pesquisa**
    - um trabalho por grupo, inclui apresentação e discussão em aula teórica

# Avaliação

## Método de avaliação

- **Época normal:**
  - [Teórica: 50%]
    - Exame: 40%, participação nas sessões de discussão: 10%
  - [Prática: 50%]
    - Mini-projectos (relatórios-35%; *log book*-5%, apresentações-10%)
- **Época de recurso:**
  - [Teórica: 50%]
    - Exame
  - [Prática: 50%]
    - Nota correspondente da época normal ou exame prático

Nota mínima de **7 valores** para qualquer das componentes

# Bibliografia

## Bibliografia principal:

- Marwedel, P., “Embedded System Design”, Springer, 2003
- Buttazzo, G., “Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications”, 2nd Ed., Springer, 2004
- Campbell, S., L., J.P. Chancelier, R. Nikoukhah, “Modeling and Simulation in Scilab/Scicos”, Springer, 2005
- Douglass, B. P., D. Harel, “Real-Time UML: Developing Efficient Objects for Embedded Systems”, 2nd Ed., Addison Wesley Longman

**Serão fornecidas referências bibliográficas adicionais em alguns módulos**