

Sistemas Digitais

Aula 10 - Sistemas Embarcados: Conceitos e Caracter sticas

Apresentação

Nesta aula, será apresentado o conceito de sistemas embarcados, bem como uma introdução aos seus fundamentos e características. Você também irá conhecer aspectos do projeto de sistemas embarcados e suas principais diferenças em relação aos computadores pessoais. Além disso, você vai conhecer algumas estratégias de projeto de um sistema embarcado que buscam acelerar esses projetos, reduzindo o tempo de lançamento do produto. Bons estudos!

Objetivos

Ao final desta aula, você será capaz de:

- Definir sistemas embarcados.
- Diferenciar sistemas embarcados de computadores convencionais.
- Estabelecer um conjunto de parâmetros de caracterização dos sistemas embarcados.
- Identificar as principais restrições dos sistemas embarcados.
- Escolher estratégias que podem acelerar os projetos de sistemas embarcados, reduzindo o tempo de lançamento do produto.

Conceito de Sistemas Embarcados

Para começar, você deve saber que não existe uma definição direta e única para sistemas embarcados. Vamos ver algumas definições para ajudar a compreender a abrangência desses sistemas no nosso cotidiano:

- Sistemas embarcados são sistemas onde *hardware* e *software*, normalmente, são integrados e seu projeto visa o desempenho de uma função específica. A palavra **embarcados** leva a ideia que esses sistemas são parte de um sistema maior. Esse sistema maior pode ser composto por outros sistemas embarcados.
- Sistemas embarcados são sistemas de processamento de informações que estão embarcados em sistemas maiores e que, normalmente, não são visíveis ao usuário.
- Um sistema embarcado é um sistema baseado em um microprocessador, que é projetado para controlar uma função ou uma gama de funções, e não para ser programado pelo usuário final, como acontece com os PCs.
- Um sistema embarcado é qualquer aparelho que possua um computador programável, mas esse não é projetado para ser um computador de uso geral.

Todas essas definições tentam descrever da forma mais ampla o que são sistemas embarcados, embora algumas dessas definições não consigam ser abrangentes o suficiente para os sistemas embarcados atuais.

Mas Afinal, o que são Sistemas Embarcados?

No nosso cotidiano temos diversos exemplos de sistemas embarcados, como vemos na Figura 1. Podemos encontrar sistemas embarcados em TV digitais, automóveis, aviões, caixas eletrônicos de banco, *tablets*, celulares e *smartphones*, além de diversos eletrodomésticos, como microondas, máquinas de lavar, geladeiras, etc. Os exemplos são muitos e não conseguiríamos listar todos aqui, mas vamos tentar encontrar uma definição que consiga definir toda essa gama de sistemas.

Sistema embarcado é todo sistema onde algum tipo de informação é processada e este possui características distintas dos sistemas computacionais convencionais, principalmente em relação a restrições mais rígidas e requisitos específicos.

Por exemplo, em uma geladeira moderna, podemos colocar um refrigerante no congelador e programá-la para deixar a bebida pronta em um determinado tempo. Os telefones celulares são outro exemplo. No início, esses aparelhos possuíam somente as funções de um telefone comum. Atualmente, além de telefones, eles também apresentam funções como câmera fotográfica, acesso à internet, telas sensíveis ao toque, tocador de música e muitas outras funções “inteligentes”, o que os tornam “*smartphones*”.

Saiba Mais!

Segundo uma pesquisa realizada pela IDC (International Data Corporation) em 2011, 19% de todos os sistemas eletrônicos vendidos hoje no mundo são sistemas embarcados, e esse número deverá crescer para 33% até 2015. Eles são, atualmente, responsáveis por uma receita anual de 1 trilhão de dólares, número que deverá dobrar no período de 4 anos, quando, então, serão responsáveis pelo consumo de 14,5 bilhões de processadores por ano, que representará mais de 2 para cada habitante do mundo.

Figura 01 - Exemplos de sistemas embarcados



Fonte: <http://studentprojectideas.in/embedded-projects-for-students/> Acesso em: 27 jul. 2012

Saiba Mais!

Segundo o Rodolfo Pellizoni, Professor Doutor da Universidade de Waterloo - Canadá, mais de 99% dos processadores produzidos na atualidade são vendidos para aplicações de sistemas embarcados.

Atividade 01

1. Pesquise na internet as cinco principais empresas fabricantes de microprocessadores e microcontroladores. Em sua pesquisa, verifique aspectos como a gama de variedade de dispositivos que cada empresa oferece e em quais aparelhos eletrônicos os *chips* de cada fabricante são utilizados.

Principais Características

Vamos analisar as principais características dos sistemas embarcados e algumas das diferenças em relação aos sistemas computacionais convencionais:

Em geral, a maioria dos sistemas embarcados são projetados para realizar uma função ou uma gama de funções e não para serem programados pelo usuário final, como os computadores pessoais. O usuário pode alterar ou configurar a maneira como o sistema se comporta, porém não pode alterar a função que esse realiza. No caso de computadores pessoais, são programados pelo usuário e podem realizar qualquer função predeterminada por ele, diferentemente dos sistemas embarcados que realizam uma ou várias funções específicas.

Essa característica aparece na grande maioria dos sistemas embarcados, mas existem também sistemas embarcados onde o usuário pode instalar novas funções ou mesmo reprogramar as existentes. Um exemplo disso, são os smartphones, onde o usuário pode instalar ou desinstalar aplicativos, modificando as funções do aparelho. Outro exemplo é quando temos sistemas embarcados com hardware reprogramável, como os **FPGAs que vimos na Aula VDHL - Parte I**, pois o circuito pode ser reprogramado para ter diferentes funcionalidades.

Nos sistemas embarcados, é comum que exista algum tipo de interação com o ambiente em que se encontram. Por exemplo, o sistema de controle do nível da água em uma máquina de lavar roupas coleta os dados do nível do tanque por meio de sensores e o modifica através de atuadores.

Sistemas embarcados devem ser confiáveis. Neles, as implicações de uma falha de *software* são muito mais preocupantes do que em um computador pessoal. Muitos desses sistemas realizam funções críticas e interagem com o ambiente ou seres humanos, como é o caso dos sistemas críticos (ou sistemas de tempo crítico), onde falhas podem causar catástrofes. Como mostra a Figura 2, um bom exemplo são os atuais carros com o sistema de acionamento dos *airbags*, onde alguns milissegundos podem fazer toda a diferença, ou o sistema de frenagem ABS, onde alguma falha pode resultar em acidentes ou até mesmo a regulação do sistema de controle na emissão de gases poluentes no ambiente. Nos casos do acionamento de *airbags* e da frenagem ABS, é dito que esses sistemas possuem requisitos de tempo crítico e são intolerantes atrasos. Outro exemplo é o *software* de controle da quantidade de radiação em uma máquina de raios X, onde uma falha poderia causar sérios danos à saúde dos pacientes. Dizer que um sistema é confiável significa que esse possui certas características de estabilidade, recuperação, disponibilidade e segurança.

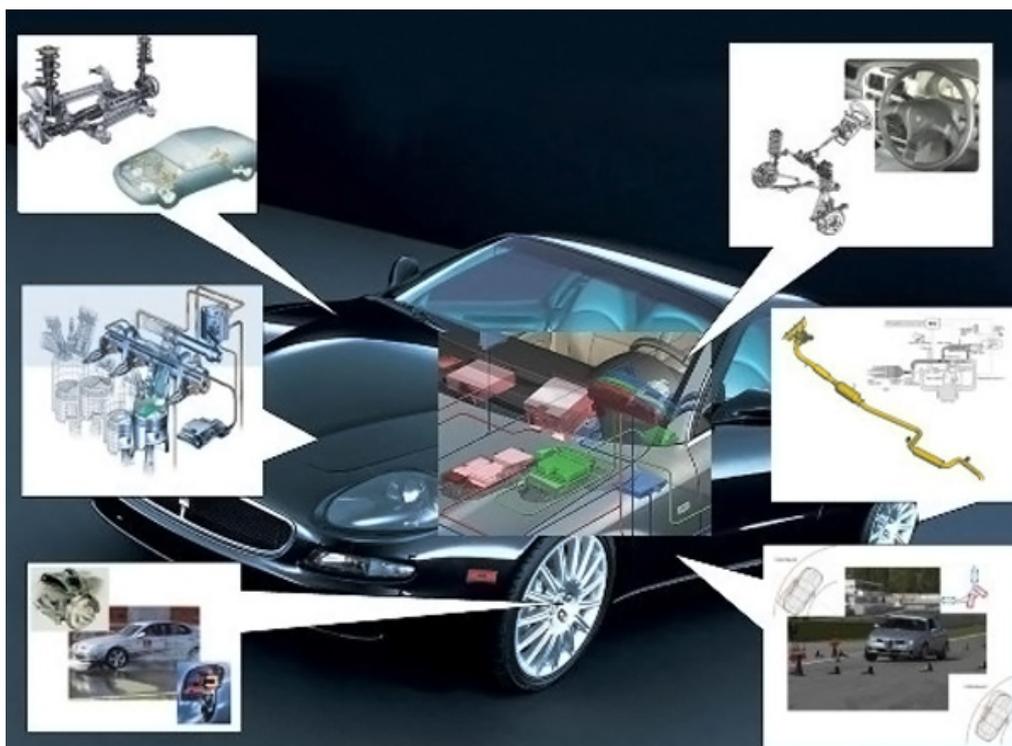
Estabilidade: é a probabilidade de que um sistema não irá falhar.

Recuperação: é a probabilidade de que uma falha no sistema será corrigida em um determinado intervalo de tempo.

Disponibilidade: é a probabilidade de que um sistema estará disponível em certo tempo. Alta estabilidade e recuperação levam a uma alta disponibilidade.

Segurança: Um sistema deve ser seguro em dois aspectos. Ele deve ser seguro para o meio ambiente, ou seja, uma falha não acarreta em danos ao meio ou às pessoas que utilizam esse sistema, e ele deve manter as informações confidenciais dentro dele, autenticando os usuários para não permitir que pessoas não autorizadas manipulem essas informações.

Figura 02 - Automóvel composto também por sistemas embarcados



Fonte: <http://embeddedsystems.onsugar.com/Embedded-System-need-able-fix-multifaceted-application-glitches-10692576> Acesso em: 27 jul. 2012.

Sistemas embarcados possuem outras métricas de eficiência, além das já conhecidas por projetistas de sistemas de computadores pessoais e servidores. Vamos estudar algumas delas:

Consumo de energia: Considerando que muitos sistemas embarcados são móveis, e são alimentados por baterias, temos que projetá-los para consumir o mínimo de energia possível. A tecnologia na fabricação de baterias evolui em ritmo muito menor que as aplicações que fazem uso dessas baterias, logo, a energia disponível deve ser aproveitada da melhor maneira possível. Afinal, não queremos ter que carregar uma bateria de carro para alimentar os nossos celulares, certo?

Área: Muitos dos sistemas embarcados que encontramos no nosso dia a dia são portáteis ou devem ocupar o menor espaço possível dentro do produto onde estão embutidos. Então, devemos sempre buscar projetar sistemas embarcados que ocupem a menor área possível.

Desempenho: Os recursos de *hardware* na maioria dos sistemas embarcados são bastante limitados. Assim, devemos projetar esses sistemas para atender os requisitos de tempo com o mínimo de recursos possível e com reduzido consumo de energia. Para reduzir o consumo de energia, a frequência do *clock* e a tensão de alimentação devem ser as menores possíveis. Então, somente devemos ter no sistema embarcado componentes de *hardware* que sejam necessários. Se, por exemplo, não for necessário realizar uma conversão de grandezas físicas (sinais analógicos) para valores digitais, não devemos utilizar conversores analógico/digital, pois isso ocuparia mais espaço e causaria um aumento do consumo de energia.

Tamanho de código: Todo código de uma aplicação que é executada em um sistema embarcado deve estar presente nele, quase sempre em memória. Muito raramente sistemas embarcados possuem dispositivos de armazenamento magnético, como discos rígidos, para armazenar código, dessa forma, a disponibilidade de memória é muito limitada, por isso, o desperdício de memória deve ser evitado pelos programadores de *software* embarcado.

Peso: Sistemas portáteis e móveis devem ser leves. Podemos comprovar isso quando comparamos os celulares de primeira geração com os celulares atuais. Sistemas móveis pesados tendem a desaparecer.

Custo: O uso eficiente de componentes de *hardware* pode baixar o custo final de um sistema embarcado. No mercado de eletrodomésticos, por exemplo, a competitividade é importantíssima e o custo é um fator decisivo para dizer se um produto possui mercado.

Interação com o usuário: Grande parte dos sistemas embarcados não possui teclados, mouse, monitores ou outros dispositivos encontrados em computadores pessoais para realizar interfaceamento com o usuário. Sistemas embarcados possuem interfaces dedicadas, como botões, *leds* e chaves. Por isso, dificilmente o usuário reconhece a informação sendo transmitida ou processada dentro deles.

Tempo real: Muitos sistemas embarcados possuem requisitos de tempo real, ou seja, precisam de previsibilidade. Não completar uma tarefa em um tempo determinado pode resultar em perda de dados e, conseqüentemente, de qualidade ou causar danos. Aplicações multimídia podem perder alguns dados com prejuízo apenas para a qualidade, mas em um sistema de controle de *airbag* em um carro ou em um sistema de controle de um forno industrial isso não pode ocorrer, pois pode

resultar em catástrofes, implicando em perdas de vida ou, no mínimo, na destruição do sistema. Sistemas de tempo real não devem utilizar componentes ou técnicas que diminuam o tempo de processamento na média, como memórias cache.

Interação com o ambiente: Tipicamente, sistemas embarcados são reativos ao ambiente, ou seja, eles estão em interação contínua com o ambiente e executam em um ritmo determinado por esse. Pode-se dizer que um sistema reativo é aquele necessita de uma entrada (interação externa, através de sensores) para reagir, dessa forma mudando seu estado atual. Para cada entrada recebida, ele realiza o processamento da informação e gera uma saída, que poderá agir no ambiente, através de atuadores.

Ao contrário dos computadores pessoais, grande parte dos sistemas embarcados é alimentada por pequenas baterias, de forma que a redução de alguns mW/h no consumo pode estender a duração da bateria por dias ou até meses. Nos sistemas embarcados, a responsabilidade de poupar energia recai tanto na engenharia do *hardware* quanto no desenvolvimento do *software*. Por exemplo, em um PDA (do inglês Personal Digital Assistants, Assistente Pessoal Digital, também conhecido como Palmtop) o consumo pode ser reduzido em até 60% por meio de alterações na maneira como o *software* é escrito. Evidentemente, as escolhas no projeto do *hardware* também têm impacto crucial no consumo de energia do sistema.

Os sistemas embarcados, normalmente, possuem uma natureza portátil, de forma que eles devem ser capazes de suportar as mesmas condições suportadas pelos seus usuários ou sistemas receptores. Por exemplo, um celular *smartphone* deve ser capaz de operar com temperaturas acima de 40°C em praias, e temperaturas abaixo de -15° em montanhas ou ainda a 11.000 metros de altitude na cabine de um avião.

Em um sistema embarcado, os recursos são mais restritos do que em computadores pessoais. Por exemplo, a memória RAM nesses sistemas, muitas vezes, não chega a 500 bytes, de forma que todo o código do *software* deve caber em alguns poucos kB. Uma simples operação matemática pode consumir 1% ou 70% da memória, dependendo do processador e da forma como a operação é escrita. A busca de tamanhos reduzidos pode ser vista na Figura 3, onde vemos um chip da ARM com tamanho de 2x2x0,5 milímetros.

Figura 03 - Um dos menores chip ARM do mundo com 2x2x0,5 milímetros.



Fonte: <http://olhardigital.uol.com.br/noticia/conheca-a-arm,-a-pedra-no-sapato-da-intel/33315>
Acesso em: 27 ago. 2013.

Atividade 02

1. Faça uma lista dos dispositivos eletrônicos que estão presentes no seu cotidiano (televisores, telefones celulares, eletrodomésticos etc.). Quais deles você acha que são sistemas embarcados? Para cada dispositivo listado, justifique o motivo de você o ter escolhido, abordando quais as características dos sistemas embarcados citadas anteriormente ele possui.

Desenvolvimento de Sistemas Embarcados

Sistemas embarcados podem ser desde sistemas bem simples, como controles remotos, até sistemas com alto grau de complexidade, com centenas de processadores em paralelo. Então, no desenvolvimento de sistemas embarcados, principalmente os mais complexos, é necessário uma metodologia para que esse desenvolvimento chegue a termo com o sistema mais adequado.

Dependendo da metodologia utilizada, as etapas podem ter uma ordem ou grau de refinamento diferentes, mas é provável que em qualquer metodologia encontremos essas etapas:

- Análise de requisitos
- Projeto
- Implementação

Análise de Requisitos

Essa é uma das etapas mais importante, pois é o momento onde fazemos o levantamento de todos os requisitos, tanto os funcionais como os não funcionais. Entre os requisitos não funcionais estão custo, peso, área, dentre outros.

Em geral, as etapas de análise de requisitos, projeto e implementação não são executadas linearmente, sendo mais adequada uma abordagem iterativa, ou seja, quando se inicia um projeto, as informações sobre os objetivos e funcionalidades são, geralmente, incompletas. Estas informações, no entanto, permitirão a obtenção de uma primeira arquitetura do sistema. Após esta fase, os objetivos e restrições obtidos na fase de análise de requisitos deverão ser revistos para complementação das informações e consequente refinamento da arquitetura.

Devemos repetir a etapa de análise de requisitos e a etapa de projeto até que a arquitetura obtida satisfaça aos objetivos e requisitos. A partir deste ponto podemos partir para a implementação. Porém, mesmo durante a implementação pode ser necessário voltarmos às etapas anteriores para complementação da funcionalidade ou satisfação de algum requisito. Um retorno neste caso pode representar um aumento significativo no custo do projeto. Quanto maior o número de iterações envolvendo as duas primeiras etapas menor será a probabilidade de um retorno a estas fases durante ou após a implementação.

Na especificação das restrições de projeto, todas as restrições internas e externas que possam comprometer o projeto devem ser verificadas. Algumas das questões, dentre muitas que devemos analisar em relação às restrições, são:

1. Qual o orçamento do projeto?
2. Quais são os prazos?

3. Quantas pessoas poderão trabalhar na equipe que desenvolverá o projeto?
4. Qual a experiência e conhecimento prévio da equipe?
5. Existe algum *hardware* ou *software* predefinido para o projeto?
6. Existe dependência de algum fornecedor?

Especialmente as duas últimas questões implicarão numa redução das possibilidades da etapa seguinte, a etapa de projeto, onde faremos uma “exploração do espaço de projeto”, avaliando todas as possibilidades que podem atender aos requisitos levantados na etapa de análise de requisitos.

Projeto

Nesta etapa devemos fazer a **exploração do espaço de projeto**, analisando as combinações possíveis na solução a ser implementada.

A demanda crescente do mercado de sistemas embarcados faz com que o tempo de projeto e lançamento do produto no mercado (*time to market*) seja cada vez menor. Além disso os produtos desse mercado possuem um tempo de vida relativamente curto em relação a outras aplicações. Então, para um produto ser competitivo no mercado deverá ter um *time to market* reduzido. Quanto maior for o atraso no lançamento do produto no mercado, maior será a perda nos lucros.

O processo de desenvolvimento tradicional de sistemas com componentes de *hardware* e de *software* baseado no desenvolvimento primeiro do *hardware* e depois do *software* tem se mostrado cada vez mais difícil e custoso. Muitas vezes, problemas na interface *hardware/software* somente são descobertos quando estamos na fase de prototipagem, ao avaliarmos o protótipo do sistema, ou até mesmo somente durante a integração e teste do sistema. A descoberta tardia desses problemas implica em projetos mais caros e em estouro do cronograma inicial, pois o reprojetado e o ajuste necessário entre *hardware* e *software* é demorado e de alto custo.

O uso de técnicas de **coprojeto hardware/software** permite uma significativa redução no custo e no tempo do projeto desses sistemas. O crescente interesse em coprojeto *hardware/software* pode ser justificado pelo avanço tecnológico e pela crescente complexidade das aplicações. A disponibilidade de ambientes de projeto suportando desde a especificação até a prototipação de sistemas digitais complexos tem permitido o projeto de uma variada gama de aplicações cada vez mais complexas.

O particionamento em *hardware* e *software* consiste em decidir como será o mapeamento do sistema entre componentes de *hardware* e *software*. Fazemos uma seleção dos componentes da arquitetura alvo e o particionamento da funcionalidade do sistema entre esses componentes. Isso deve atender restrições de projeto tais como: custo, desempenho, área e consumo de energia.

Outra estratégia bastante usada para ajudar na redução do tempo e do custo de projeto é o **reuso** dos componentes de *hardware* e de *software* preexistentes. Esses componentes podem já ter sido desenvolvidos pela equipe ou podem ser licenciados de terceiros.

Em muitos projetos também conseguimos reduzir o "*time to market*" quando fazemos o reuso de uma plataforma. Uma plataforma representa um conjunto de componentes fixos de *hardware* e de *software*, cuja complexidade de desenvolvimento implicaria em um tempo maior de projeto. Assim, um **projeto baseado em plataforma** permite que avaliemos componentes configuráveis de *hardware* e de *software* associados a essa plataforma composta pelos componentes complexos. Fazemos isso com a intenção de reduzir custos e tempo de desenvolvimento, riscos do projeto, além do *time to market*.

Saiba Mais!

Alguns exemplos de plataformas muito utilizadas em projetos baseados em plataforma são: FPGAs, Arduíno e Rapsberry PI.

Se você quiser conhecer melhor as plataformas Arduíno e Rapsberry PI ou outras plataformas que possam ser utilizadas em projetos baseados em plataforma, pode fazer uma pesquisa na internet. Acesse os links para melhor esclarecimento:

<https://www.arduino.cc/>

<https://www.raspberrypi.org/>

Implementação

Na etapa de implementação devemos seguir o que foi definido nas etapas anteriores de análise de requisitos e de projeto.

Fazemos, então, a integração do componentes e os testes e validações necessários. Caso os resultados não sejam satisfatórios, devemos retornar às etapas anteriores e realizar os ajustes necessários até que os objetivos e requisitos sejam atendidos. Esses retornos podem causar um aumento significativo no custo do projeto.

Atividade 03

1. Faça uma pesquisa a respeito dos requisitos não funcionais de um sistema embarcado. Cite quais são e explique cada um deles.

Tendências em Sistemas Embarcados

Sistemas embarcados estão presentes cada vez mais no nosso cotidiano, mas muitas vezes não percebemos (e pode ser que não cheguemos a perceber) todos os sistemas com os quais interagimos.

Os carros têm cada vez mais eletrônica embarcada, desde sistema de controle de *airbag*, de freios abs até sistemas de entretenimento, como DVDs e tocadores de MP3. O número de processadores veiculares é cada vez maior e alguns estão conectados em rede. Essas redes veiculares, também conhecidas como VANETs podem interligar esses processadores, como também chegar a um controle mais efetivo como os sistemas de automação veicular, permitindo o deslocamento e controle do carro sem a interferência do motorista.

Wearable computers (ou computadores “vestíveis”), como o Google Glass (óculos do Google), permitem que possamos interagir com os sistemas computacionais embarcados com uma nova perspectiva de interação, sem teclados, mouses ou telas.

Essa possibilidade de interação cada vez menos perceptível também atinge a comunicação entre os próprios sistemas. Será cada vez mais comum termos diversos equipamentos “conversando” entre si. Carros “conversando” com outros carros, com semáforos e com outras “coisas” representam os cenários de **Cidades inteligentes**, onde a **Internet das coisas** permitirá um cenário de computação ubíqua.

Atividade 04

1. Pesquise sobre os processadores usados em *tablets* e *smartphones* e descubra qual o fabricante que domina esse segmento.
2. Pesquise sobre outras projeções futuras em relação ao uso de sistemas embarcados.

Resumo

Nesta aula, você conheceu algumas definições de sistemas embarcados e teve a oportunidade de observar o quanto eles estão presentes em nosso cotidiano. Além disso, aprendeu as principais características desses sistemas e como se diferenciam de sistemas computacionais convencionais. Para finalizar, você também aprendeu que em um projeto de sistemas embarcados os projetistas devem seguir algumas regras estabelecidas.

Autoavaliação

1. Depois do que estudamos, defina com suas próprias palavras o que são sistemas embarcados, procurando abordar os aspectos principais que vimos nesta aula.
2. Dê 10 exemplos de dispositivos que podem ser classificados como sistemas embarcados que fazem parte do nosso cotidiano. Justifique cada uma de suas escolhas.
3. Cite e explique 3 diferenças dos sistemas embarcados em relação à sistemas computacionais convencionais.
4. Sistemas embarcados precisam ser confiáveis. Essa afirmação é verdadeira? Explique com suas palavras e utilize exemplos. Além disso, quais são as características de um sistema confiável? Explique cada uma dessas características.
5. Perspectivas de futuro (ou já de presente) para sistemas embarcados apontam para o uso deles em ambientes de computação ubíqua e pervasiva . A internet das coisas no cenário de casas e cidades inteligentes é um exemplo disso. Busque exemplos de sistemas embarcados que são usados nesse contexto e em outros cenários futuros previstos.

Referências

BERGER, A. S. **Embedded Systems Design**: an Introduction to Process, Tools, & Techniques. USA: CMP Books, 2002.

HEATH, Steve. **Embedded System Design**. 2. ed. São Paulo: Elsevier, 2003.

MARWEDEL, P. **Embedded System Design**. USA: Kluwer Academic Publishers, Dortmund, 2003.

SANTOS, Danillo Moura. **Projeto de sistemas embarcados**: um estudo de caso baseado em microcontrolador e seguindo

AOSD. 2005. Disponível em:
<http://www.lisha.ufsc.br/pub/Santos_BSC_2005.pdf>. Acesso em: 10 abr. 2012.