

TEMA 08

SmartThings

Habilidades

- Habilidade em integrar **dispositivos inteligentes** (SmartThings) em sistemas.

8.1. Definição

O crescimento da indústria impulsiona o avanço das tecnologias eletrônicas, fato este ilustrado pela adoção massiva dos smartthings em dispositivos eletrônicos nos diversos itens produzidos, como automóveis, eletrodomésticos, equipamentos de comunicação pessoal, dentre outros. Atualmente o termo sistema embarcado ganhou notoriedade com o surgimento das tecnologias móveis e principalmente a convergência dessas tecnologias as plataformas de prototipação possibilitando o desenvolvimento do conceito em torno da Internet das Coisas (do inglês: Internet of Things ou IoT).

Aqui estão alguns exemplos de sistemas embarcados:

1. ****Eletrônicos de Consumo:****

- Smartphones: Eles contêm uma variedade de sistemas embarcados para gerenciar funções como comunicação sem fio, sensores, câmeras e processamento de dados.
- Smartwatches e dispositivos de rastreamento de fitness: Eles incluem sistemas embarcados para monitorar atividades físicas, frequência cardíaca e outras métricas de saúde.
- TVs inteligentes: Integradas com sistemas embarcados para permitir streaming de vídeo, interatividade com aplicativos e controle remoto.

2. ****Automóveis:****

- Sistemas de controle do motor e transmissão: Eles usam sistemas embarcados para otimizar o desempenho do veículo, gerenciar consumo de combustível e reduzir as emissões.
- Sistemas de entretenimento e navegação: Presentes em muitos carros modernos, esses sistemas embarcados oferecem funcionalidades como GPS, reprodução de mídia e conectividade Bluetooth.
- Sistemas de segurança: Como os sistemas de freios antibloqueio (ABS) e airbags, que dependem de sensores e sistemas embarcados para detectar situações de perigo e reagir rapidamente.

3. ****Eletrodomésticos:****

- Geladeiras inteligentes: Equipadas com sistemas embarcados para controlar a temperatura, monitorar estoques de alimentos e até mesmo fazer pedidos de compras automaticamente.
- Máquinas de lavar e secar roupa: Possuem sistemas embarcados para gerenciar ciclos de lavagem, controlar a temperatura e garantir eficiência energética.
- Fornos e fogões: Incorporam sistemas embarcados para controlar a temperatura, gerenciar o tempo de cozimento e oferecer recursos de segurança, como desligamento automático.

4. ****Dispositivos Médicos:****

- Monitores de glicose: Utilizam sistemas embarcados para medir os níveis de glicose no sangue e fornecer informações em tempo real para os pacientes e médicos.
- Desfibriladores automáticos: Equipados com sistemas embarcados para analisar o ritmo cardíaco e administrar choques quando necessário.

5. ****Sistemas de Controle Industrial:****

- Automação industrial: Muitas fábricas e instalações de produção dependem de sistemas embarcados para controlar máquinas, monitorar processos e otimizar a produção.
- Sistemas de monitoramento e controle ambiental: Presentes em edifícios comerciais e residenciais para regular a temperatura, a iluminação e o consumo de energia.

Esses são apenas alguns exemplos, mas os sistemas embarcados estão realmente em todos os lugares, desde dispositivos simples até sistemas complexos e críticos para a segurança e o funcionamento eficiente de uma ampla variedade de tecnologias e indústrias.

Para aprofundarmos no estudo de smartthings é necessário ter um conhecimento em hardware, software e em interface homem-máquina. A diferença entre o computador e o sistema embarcado é a quantidade de execução de tarefas, os computadores funcionam com sistemas operacionais e softwares com diversas aplicações, e o **sistema embarcado é dedicado para executar uma tarefa pré-determinada**.

Todas as ações realizadas pelo sistema embarcado são controladas por um microprocessador ou microcontrolador que tem a função de um “cérebro”, assim lendo os sinais externos de sensores, executando a programação, processando os sinais e enviando para os atuadores.

O diagrama básico de um sistema embarcado com microcontrolador, tem a memória com a função de armazenar as variáveis da programação e armazenar a própria programação, os periféricos que tem a função de auxiliar na aplicação, os conversores A/D (Analogico/Digital) que convertem sinais analógicos recebidos dos sensores em sinais digitais, os conversores D/A (Digital/Analógico) que convertem sinais digitais em analógicos para controlar atuadores que funcionam de modo analógico, seja linear (proporcional) ou não-linear.

O projetista de sistema embarcado tem a noção de programação, de controle de processos, de dispositivos de aquisição de dados, de conversores, de sensores, de atuadores e de estruturação do projeto. Além de ter esses conhecimentos necessários, é preciso sempre estar atualizado nos lançamentos de novas tecnologias que podem ser utilizadas nos smartthings e IoT.

8.2. Histórico

O Apollo Guidance Computer (AGC) foi o **primeiro sistema embarcado da história**, desenvolvido pelo MIT (Instituto de Tecnologia de Massachusetts) no início dos anos 60 para equipar as naves do projeto Apollo, já o primeiro sistema embarcado de produção em massa foi o computador guia do míssil nuclear LGM-30 Míssil Minuteman, lançado em 1961. Ele possuía um disco rígido para a memória principal. Quando a segunda versão do míssil entrou em produção em 1966, o computador guia foi substituído por um novo, que constituiu o primeiro uso em grande volume de circuitos integrados (NETO et al., 2016; NEVES et al., 2013).

Desde suas primeiras aplicações na década de 1960, os smartthings vêm reduzindo seu preço. Também tem havido um aumento no poder de processamento e funcionalidade. **Em 1978 foi lançada pela National Electrical Manufacturers Association (NEMA) a norma para microcontroladores programáveis (FERNANDES et al., 2017)**.

Em meados da década de 1980, vários componentes externos foram integrados no mesmo chip do processador, o que resultou em circuitos integrados chamados microcontroladores e na difusão dos SmartThings. Com o custo dos microcontroladores menor que um dólar americano, tornou-se viável substituir componentes analógicos caros, como potenciômetros e capacitores por eletrônica digital controlada por pequenos microcontroladores.

8.3. Plataformas para SmartThings

Existem diversas plataformas de smartthings, atualmente as plataformas mais utilizadas para fins acadêmico e profissional são o **Arduino e o Raspberry Pi**. Na área acadêmica os alunos utilizam as plataformas de smartthings para desenvolver projetos (protótipos) automatizados, principalmente para desenvolver trabalhos finais de curso, já no meio profissional é utilizado em projetos que têm produção em massa, depois de testado e aprovado por especialistas. Seguem as duas plataformas usuais:

Arduino:

O Arduino é parte integrante do conceito de hardware e software livre, surgiu sob o propósito de impulsionar a prototipação de maneira descomplicada tanto para especialistas em eletrônica como também em software. Seu projeto original foi criado em 2005 por um grupo de 5 (cinco) pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. O objetivo era elaborar um dispositivo com um bom custo-benefício, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores (RITO, 2017).

Um ponto importante no protótipo embarcado Arduino é a utilização de “socket” para o encaixe do microcontrolador, assim possibilita a troca do microcontrolador, caso necessite.

Os dispositivos de baixo custo, como o Arduino e outros controladores, são projetados em **tecnologia TTL (Transistor–Transistor Logic) com níveis de tensão específicos**. O nível alto 1 (binário) é definido como aceitável dentro da escala; o nível indefinido pode ser alto ou baixo, sem previsão; e o nível baixo pode ser considerado 0 (binário) dentro da escala (FELIX, 2018).

Uma vantagem do controlador Arduino é a possibilidade de utilizar conversores AD/DA (Analógico-Digitais/Digitais-Analógicos) embutidos no microcontrolador, **não necessitando de placas ou recursos externos para realizar leituras de sinais analógicos ou controlar atuadores proporcionais com sinais analógicos**. Porém, esses conversores têm uma limitação na resolução (em bits) de conversão. Caso necessite de maiores precisões existem no mercado outros conversores externos que podem ter resoluções maiores.

A plataforma Arduino é muito simplificada e fácil de ser acessada, pois seu custo está bem acessível no mercado. Ao adquirirmos uma placa Arduino é preciso somente “plugar” os cabos nas portas de comunicação e realizar algumas configurações básicas, depois somente desenvolver os projetos.

A plataforma **Raspberry Pi vem trazendo avanços em termos de processamento e performance em projetos**. A principal vantagem do protótipo Raspberry Pi é **realizar a aquisição de dados das variáveis de medida em tempo real**, pois apresenta uma alta performance (FELIX et al., 2017).

O **Raspberry Pi possui uma desvantagem relacionada às variáveis analógicas, pois possui somente entradas e saídas digitais**. Para realizar a aquisição de variáveis analógicas é necessária uma placa eletrônica externa com conversores AD/DA (Analógico - Digital e Digital - Analógico) (FELIX et al., 2017). Outra desvantagem do controlador Raspberry Pi em comparação ao Arduino é o custo, pois o Raspberry Pi possui um preço elevado em relação ao Arduino, mas tem em destaque a melhor performance em processamento.

A placa de conversores AD/DA possui conversores AD de 24 bits de resolução e conversores DA de 16 bits de resolução. A arquitetura básica do conversor DA utiliza o chipset DAC8532 junto com um registrador para guardar os valores digitais recebidos e convertê-los em níveis de tensão variáveis; um “resistor string” que realiza a divisão da tensão de referência; e um amplificador com a finalidade de gerar, na saída, a tensão de 0 a 5 V. A arquitetura básica do conversor AD utiliza o chipset ADS1256 e possui as seguintes características: seleção de entradas com multiplex; sensor para detectar a entrada utilizada; amplificador de ganho programável (PGA) para modificar a quantidade de amostra, por segundo, com o padrão de 30k sps (samples per second); é um modulador de quarta ordem juntamente com um filtro digital programável para configurar o PGA. A comunicação compatível utilizada é a SPI (Serial Peripheral Interface) para controlar os conversores por meio do controlador Raspberry Pi (FELIX et al., 2017).

Prototipação

A prototipação em sistema embarcado pode ser caracterizada como componentes eletrônicos montados em uma ou várias placas que compõem o sistema embarcado.

É essencial desenvolver os protótipos de smartthings na matriz de contato antes de confeccioná-los em circuitos impressos, assim poderá verificar se o circuito eletrônico desenvolvido está funcionando corretamente.

A prototipação não se restringe apenas aos circuitos eletrônicos do sistema embarcado, assim, podendo ser desenvolvidos circuitos auxiliares (módulos ou “shields”) que funcionam junto com as placas de smartthings, por exemplo, módulo relé, módulo WiFi, módulo de infravermelho e outros. Atualmente, os simuladores são utilizados para verificar o funcionamento do circuito eletrônico, antes de montar o protótipo na matriz de contato. Existem simuladores específicos para desenvolver circuitos de smartthings, hoje os mais utilizados são: Tinkercad, UnoArduSim, Fritzing e outros.

8.4.Introdução à Linguagens de Programação

A utilização da linguagem de programação desempenha um papel fundamental no desenvolvimento da lógica por trás das aplicações que operam com dispositivos SmartThings. A programação de dispositivos SmartThings é crucial para estabelecer uma comunicação efetiva entre o protótipo embarcado e as interfaces externas, permitindo a execução de um conjunto de comandos de acordo com os requisitos específicos do projeto embarcado. Portanto, **ao criar um código de programação, é fundamental identificar a linguagem de programação apropriada a ser empregada**, visto que essa linguagem é essencialmente o "idioma" utilizado para comunicar-se com o dispositivo.

Por outro lado, um algoritmo é caracterizado como uma sequência bem definida de instruções destinadas a executar uma função específica. Um exemplo de algoritmo fora do contexto da computação pode ser encontrado em uma simples "receita de bolo", na qual cada passo deve ser seguido meticulosamente para garantir que o bolo seja preparado corretamente. A maneira como um programa de computador (ou seja, um algoritmo) é elaborado é determinada pela lógica de programação subjacente, que orienta a estruturação e a ordem das instruções a serem seguidas.

Atualmente existem duas linguagens de programação populares, a linguagem C e a linguagem Python.

1. **Linguagem C e Compilação:**

- É verdade que a linguagem C é compilada, o que significa que o código-fonte escrito em C é traduzido para linguagem de máquina pelo compilador e depois executado pelo sistema operacional. O compilador C gera arquivos binários executáveis que contêm instruções em linguagem de máquina específica para a arquitetura do processador alvo, mas esses arquivos executáveis podem incluir várias seções, como código, dados, e símbolos para facilitar a execução do programa.

- A vantagem mencionada de o cliente do projeto não ver o código-fonte é verdadeira, pois os arquivos binários compilados não contêm o código-fonte original, tornando-o **menos acessível para usuários finais**.

2. **Linguagem Python e Interpretação:**

- Python é de fato uma linguagem interpretada, o que significa que o código-fonte é executado linha por linha pelo interpretador Python, que converte as instruções em linguagem de máquina em tempo de execução.

- No entanto, o **processo de interpretação não necessariamente ocorre "linha por linha"**. O interpretador pode usar técnicas como compilação just-in-time (JIT) para otimizar a execução do código Python.

- **A afirmação sobre o recuo no Python está correta**. Em Python, a indentação é fundamental para determinar a estrutura do código, como blocos condicionais, loops e definição de funções. Isso contrasta com o uso de chaves `{}` em linguagens como C para delimitar blocos de código.

ATIVIDADES:

1. O que são smartthings e como eles são utilizados em dispositivos eletrônicos?
2. Qual é a importância do conhecimento em hardware, software e interface homem-máquina para o estudo de smartthings?
3. Qual é a diferença fundamental entre um computador convencional e um sistema embarcado?
4. Qual é a função de um microprocessador ou microcontrolador em um sistema embarcado?
5. Como a evolução da tecnologia móvel e a convergência de tecnologias impactam o desenvolvimento de sistemas embarcados e o conceito de Internet das Coisas (IoT)?
6. Quais são os componentes básicos de um sistema embarcado com microcontrolador?
Explique a função de cada um deles.
7. Quais são as principais plataformas utilizadas para desenvolver smartthings e projetos de IoT?
Como elas são aplicadas em contextos acadêmicos e profissionais?
8. Explique a relação entre o Apollo Guidance Computer (AGC) e os primeiros sistemas embarcados.
Como esses sistemas evoluíram ao longo do tempo?
9. Quais são as vantagens e desvantagens das plataformas Arduino e Raspberry Pi para o desenvolvimento de smartthings?
10. Detalhe a arquitetura e as características dos conversores AD/DA presentes no Raspberry Pi.
Como esses conversores contribuem para a aquisição e controle de variáveis analógicas?

RESPOSTAS:

1. Smartthings são dispositivos eletrônicos que integram tecnologia embarcada para executar funções específicas ou automatizar processos. Eles são utilizados em uma variedade de dispositivos eletrônicos, como smartphones, eletrodomésticos, veículos, dispositivos médicos e sistemas industriais, para proporcionar funcionalidades inteligentes, monitoramento, controle e automação de tarefas.
2. O conhecimento em hardware é importante para entender a arquitetura e os componentes físicos dos sistemas embarcados, como microprocessadores, sensores e atuadores. O conhecimento em software é essencial para desenvolver programas e algoritmos que controlam o comportamento dos dispositivos. Além disso, a compreensão da interface homem-máquina é crucial para projetar sistemas que ofereçam interação intuitiva e eficiente com os usuários, garantindo uma experiência satisfatória.
3. A diferença fundamental entre um computador convencional e um sistema embarcado é a finalidade e o escopo de execução de tarefas. Enquanto os computadores convencionais são projetados para realizar uma variedade de tarefas com um sistema operacional completo e softwares diversos, os sistemas embarcados são dedicados a executar uma função específica ou um conjunto limitado de tarefas, muitas vezes sem um sistema operacional completo.
4. Um microprocessador ou microcontrolador em um sistema embarcado atua como o "cérebro" do dispositivo, sendo responsável por controlar todas as operações do sistema. Ele lê os sinais dos sensores externos, executa o programa pré-programado, processa os dados e envia comandos para os atuadores, coordenando assim o funcionamento do dispositivo.
5. A evolução da tecnologia móvel e a convergência de tecnologias têm impactado o desenvolvimento de sistemas embarcados e o conceito de IoT ao possibilitar a miniaturização, a conectividade sem fio e o aumento do poder de processamento dos dispositivos. Isso permite a criação de dispositivos mais inteligentes, interconectados e capazes de coletar, processar e compartilhar dados em tempo real, impulsionando assim a adoção e a expansão da Internet das Coisas.

6. Os componentes básicos de um sistema embarcado com microcontrolador incluem:

- Memória: Armazena variáveis da programação e o próprio código do programa.
- Periféricos: Auxiliam na interação do sistema com o ambiente externo.
- Conversores A/D (Analógico/Digital): Convertem sinais analógicos dos sensores em sinais digitais.
- Conversores D/A (Digital/Analógico): Convertem sinais digitais em sinais analógicos para controlar atuadores.

7. As principais plataformas utilizadas para desenvolver smartthings e projetos de IoT são o Arduino e o Raspberry Pi. No contexto acadêmico, os alunos utilizam essas plataformas para desenvolver projetos de automação e controle como parte de seus trabalhos finais de curso. No meio profissional, elas são empregadas em projetos de produção em massa após serem testadas e aprovadas por especialistas.

8. O Apollo Guidance Computer (AGC) foi o primeiro sistema embarcado da história, desenvolvido para as naves do projeto Apollo nos anos 60. Desde então, os sistemas embarcados evoluíram em termos de redução de custos, aumento de poder de processamento e funcionalidades, além de se tornarem mais acessíveis e difundidos em uma ampla gama de dispositivos e indústrias.

9. As vantagens do Arduino incluem seu baixo custo, simplicidade de uso e capacidade de integração de conversores AD/DA embutidos. Já as vantagens do Raspberry Pi incluem sua alta performance de processamento e capacidade de aquisição de dados em tempo real. As desvantagens do Arduino incluem sua limitação em termos de poder de processamento e recursos analógicos integrados, enquanto as desvantagens do Raspberry Pi incluem seu preço mais elevado e a necessidade de hardware externo para lidar com entradas e saídas analógicas.

10. O Raspberry Pi não possui conversores AD/DA integrados em sua placa. Para lidar com sinais analógicos, é necessário usar hardware externo, como módulos de conversores AD/DA. Esses módulos variam em características como resolução, taxa de amostragem e precisão. Eles permitem ao Raspberry Pi converter sinais analógicos em digitais para processamento e gerar sinais analógicos para controlar atuadores, facilitando o desenvolvimento de sistemas embarcados e projetos de IoT que requerem interação com variáveis analógicas.