MONITORAMENTO DE TEMPERATURA EM PYTHON

JOÃO VICTOR BATISTA DE OLIVEIRA

UNIC: Universidade de Cuiabá.

Introdução

O monitoramento contínuo de variáveis ambientais é um componente essencial em sistemas embarcados modernos, abrangendo desde aplicações industriais até projetos educacionais. A temperatura, em particular, representa um parâmetro crítico para supervisão de equipamentos, prevenção de falhas e controle de processos. Tradicionalmente, esses sistemas envolvem sensores físicos conectados a microcontroladores, que transmitem dados para um servidor central. No entanto, limitações técnicas ou de hardware podem exigir soluções alternativas, capazes de simular o comportamento de dispositivos reais em ambientes exclusivamente digitais. Neste contexto, o presente projeto propõe o desenvolvimento de uma aplicação cliente-servidor totalmente digital, com monitoramento de temperatura em tempo real, banco de dados local e interface gráfica dinâmica acessível via navegador.

Objetivo

O objetivo central deste projeto é construir um sistema embarcado digital que simula o funcionamento completo de uma arquitetura de monitoramento ambiental baseada em temperatura. Pretende-se desenvolver um modelo cliente-servidor capaz de: (1) receber dados de temperatura provenientes de um cliente externo ou de um cliente digital interno; (2) armazenar valores historicamente em um banco de dados SQLite; (3) apresentar visualizações em tempo real por meio de um dashboard web interativo.

Além disso, este projeto integra o processo avaliativo da disciplina de Sistemas Embarcados, ministrada pelo professor Felipe Douglas, como parte da avaliação do segundo bimestre do curso de Ciência da Computação da Universidade de Cuiabá (UNIC). Assim, a proposta também visa demonstrar a aplicação prática de conceitos teóricos abordados em sala, reforçando a compreensão dos estudantes sobre arquiteturas embarcadas, comunicação cliente-servidor e sistemas de monitoramento em tempo real.

Metodologia

O desenvolvimento do sistema foi estruturado em três camadas principais: servidor, cliente e

REFERÊNCIAS:

https://ojs3.perspectivasonline.com.br/exatas_e_engenharia/article/view/1599?utm_.com

https://www.sodebras.com.br

https://www.repositorio.ufal.br/handle/riufal/5120

JOÃO VICTOR BATISTA DE OLIVEIRA

UNIC: Universidade de Cuiabá.

visualização. A camada de servidor foi implementada utilizando o framework Flask em Python, responsável por gerenciar as rotas HTTP e operar como núcleo da arquitetura cliente-servidor. O servidor também realiza a persistência de dados em um banco SQLite, criado automaticamente na inicialização e composto pelos campos de temperatura e timestamp, garantindo a rastreabilidade histórica das medições simuladas.

Para a camada de aquisição de dados, foi adotada uma solução dupla. Primeiramente, um cliente externo em Python pode enviar leituras por meio de requisições POST, permitindo testes que simulam dispositivos embarcados tradicionais. Em paralelo, implementou-se um cliente digital interno utilizando threads, capaz de gerar automaticamente valores de temperatura em intervalos regulares. Essa abordagem assegura a operação contínua do sistema mesmo na ausência de um microcontrolador real, mantendo o fluxo de dados ativo tanto localmente quanto em ambiente de hospedagem.

A camada de visualização foi implementada através de um dashboard web desenvolvido com HTML, Bootstrap e a biblioteca Chart.js. Essa interface faz requisições periódicas ao servidor, atualizando o gráfico em tempo real e exibindo alertas quando limites críticos de temperatura são ultrapassados. Por fim, foi adotado um modelo híbrido de inicialização do servidor, capaz de detectar se a aplicação está sendo executada localmente ou via serviço em nuvem, ajustando automaticamente os parâmetros de execução para cada ambiente.

Resultado

O sistema final apresentou um funcionamento completo e estável, reproduzindo com fidelidade o comportamento típico de um sistema embarcado com monitoramento contínuo. A integração entre servidor Flask, banco SQLite e dashboard resultou em uma solução dinâmica e responsiva, com atualização automática do gráfico e registro consistente dos dados em banco. O cliente digital interno mostrou-se eficaz ao garantir a geração contínua de valores, permitindo que o

REFERÊNCIAS:

https://ojs3.perspectivasonline.com.br/exatas_e_engenharia/article/view/1599?utm_.com

https://www.sodebras.com.br

https://www.repositorio.ufal.br/handle/riufal/5120

JOÃO VICTOR BATISTA DE OLIVEIRA

UNIC: Universidade de Cuiabá.

dashboard opere mesmo sem a execução do cliente externo.

A interface web demonstrou boa usabilidade, apresentando valores de forma clara e destacando visualmente temperaturas acima do limite crítico configurado. O sistema mostrou-se flexível, podendo ser executado localmente ou configurado para deploy, mantendo o mesmo comportamento funcional. A organização modular permitiu ainda que componentes como o cliente externo ou o cliente digital interno fossem ativados ou desativados conforme a necessidade, aumentando a adaptabilidade do projeto.

Conclusão

O projeto alcançou plenamente seus objetivos ao criar um sistema embarcado digital funcional, capaz de simular operações típicas de monitoramento de temperatura presentes em aplicações reais. A arquitetura desenvolvida demonstrou que soluções exclusivamente digitais podem replicar de maneira eficiente o ciclo completo de coleta, tratamento, armazenamento e exibição de dados, mesmo na ausência de hardware físico. O servidor Flask, aliado ao banco SQLite e ao dashboard em tempo real, constituiu um ecossistema robusto e escalável, enquanto o cliente digital interno assegurou autonomia e continuidade ao sistema.

A experiência reforça a importância de arquiteturas cliente-servidor no desenvolvimento de sistemas embarcados modernos e evidencia o potencial de simulações digitais como ferramenta pedagógica e de prototipação. Em suma, o trabalho oferece uma base sólida para futuras expansões, incluindo integração com sensores reais, dispositivos IoT ou hospedagem em nuvem para monitoramento distribuído.

REFERÊNCIAS:

https://ojs3.perspectivasonline.com.br/exatas_e_engenharia/article/view/1599?utm_.com

https://www.sodebras.com.br

https://www.repositorio.ufal.br/handle/riufal/5120