

## 15º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2024

### Estudo de um Sistema Operacional de Tempo Real Aplicado a um Sistema Embarcado de Controle de Temperatura

**RESUMO:** Este trabalho apresenta o uso de um Sistema Operacional de Tempo Real (*RTOS*) aplicado a um sistema embarcado para controle de temperatura. Durante a implementação, foram analisados aspectos como a complexidade no desenvolvimento de *software*, a gestão eficiente de tarefas em tempo real e a sincronização precisa de eventos. Foram explorados os benefícios do uso de um *RTOS*, desde a simplificação do desenvolvimento até a otimização do gerenciamento de recursos e do desempenho do sistema de controle térmico. Os resultados demonstraram a validação do funcionamento de um circuito eletrônico para uma aplicação de controle de temperatura utilizando o *FreeRTOS* em um microcontrolador *ARM* (*STM32F401*), oferecendo uma base para a criação de materiais que possam apoiar o desenvolvimento de novos projetos.

**PALAVRAS-CHAVE:** Controle de Temperatura, *FreeRTOS*, Microcontrolador *ARM*

**Study of a Real-Time Operating System Applied to an Embedded Temperature Control System**

**ABSTRACT:** This work presents the use of a Real-time Operating System (RTOS) applied to an embedded system for temperature control. Throughout the implementation, aspects were analyzed, such as software development complexity, efficient task management in real-time, and precise event synchronization. The benefits of using an RTOS were explored, from simplifying the process development to optimizing resource management and system performance in thermal control. Results demonstrated the electronic circuit validation for temperature control employing the FreeRTOS in an ARM microcontroller (STM32F401), paving the way for creating materials to support future projects development.

**KEYWORDS:** *Temperature Control, FreeRTOS, ARM Microcontroller.*

### INTRODUÇÃO

Projetos que utilizam microcontroladores ou microprocessadores têm se tornado cada vez mais comuns devido à facilidade de acesso a esses dispositivos e à ampla gama de aplicações em que podem ser utilizados. Tais aplicações, conhecidas como sistemas embarcados, consistem em um conjunto de *hardware* e *software* projetados para executar uma tarefa específica dentro de um sistema maior. Segundo Souza (2022), esses sistemas visam realizar o controle ou monitoramento de determinadas aplicações e são projetados para atender a requisitos como baixo custo, baixo consumo de energia, compactade e confiabilidade.

O aumento na complexidade dessas aplicações de Internet das Coisas (IoT) e de outras áreas, como instrumentação e controle de processos, traz desafios significativos no desenvolvimento de *hardware* e *firmware*. Para enfrentar esses desafios, métodos não tradicionais de programação, como o

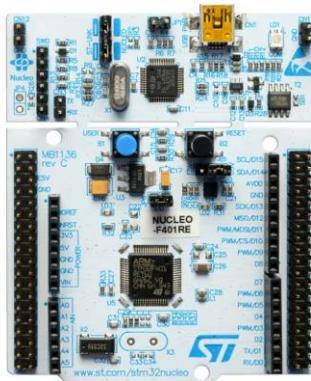
uso de Sistemas Operacionais de Tempo Real (RTOS), podem ser extremamente úteis. Segundo Morais (2018), um RTOS oferece um ambiente multitarefa que proporciona uma abstração entre *firmware* e *hardware*, além de incluir o gerenciamento de recursos internos, essencial para garantir o tempo de resposta das atividades em sistemas críticos.

O objetivo deste trabalho consiste no estudo de uma aplicação de controle de temperatura proposto em Bastos (2021), utilizando como base o Sistema Operacional de Tempo Real *FreeRTOS* (AMAZON WEB SERVICES, 2024) embarcado em um microcontrolador STM32. A pesquisa visa gerar conhecimentos e materiais que possam ser utilizados como referência para o desenvolvimento de novos projetos na área de sistemas embarcados. Dessa forma, espera-se contribuir para a inovação e a eficiência em futuras aplicações que requerem controle preciso e confiável.

## MATERIAL E MÉTODOS

Para que seja possível validar o projeto como um todo (*hardware* e *firmware*), foram realizados treinamentos com as ferramentas de desenvolvimento da empresa STMicroelectronics: a placa Nucleo-F401RE (Figura 1) e o software de desenvolvimento integrado STM32CubeIDE.

Figura 1: Ferramentas de desenvolvimento da empresa *STMicroelectronics*.



Fonte: *STMicroelectronics*,2024<sup>6</sup>.

Após a fase inicial de estudo com as ferramentas mencionadas, investigaram-se os métodos de integração e programação do *FreeRTOS* na plataforma, permitindo a aplicação dos conceitos básicos de um *RTOS*.

De acordo com Kopetz (2011), tais conceitos envolvem a execução de tarefas baseadas em um escalonador (*scheduler*) que gerencia o tempo de execução de cada tarefa de acordo com um período definido. As tarefas podem ter diferentes níveis de prioridade, o que determina a ordem em que são executadas pelo sistema. Quando uma tarefa é interrompida, seu contexto (informações necessárias para retomar a execução) é salvo na pilha (*stack*). Cada tarefa possui um laço infinito, no qual realiza as suas operações, aguardando ser reativada pelo escalonador.

O conceito de filas (*queues*) foi utilizado para a comunicação eficiente entre as tarefas (*tasks*) no sistema. Diferentemente dos semáforos, que servem para sinalizar eventos ou condições, as filas permitem a transferência de dados de uma tarefa para outra de forma segura e controlada. No código, a fila “*adcQueue*” foi criada utilizando a função “*xQueueCreate*”, que aloca espaço para armazenar até 10 valores de 16 bits. Essa fila é usada para enviar os valores lidos pelo Conversor Analógico-Digital (ADC) em uma tarefa, para outra tarefa, que é responsável pelo cálculo da média móvel e transmissão dos valores para o computador por meio de uma comunicação serial.

A criação de tarefas no *FreeRTOS* segue a sintaxe da função “*xTaskCreate*” que define a função a ser executada pela tarefa, seu nome, o tamanho da pilha em *words*, parâmetros de entrada (caso existam), a prioridade da tarefa, e um ponteiro para armazenar o identificador (*handle*) da tarefa (caso necessário). No exemplo abaixo, duas tarefas foram criadas: *vMovingAverage* para calcular a média móvel dos valores de ADC e *vReadADC* para realizar a leitura do ADC:

<sup>6</sup> Disponível em: <<https://www.st.com/en/evaluation-tools/nucleo-f401re.html>>. Acesso em: 24 de Julho 2024

```

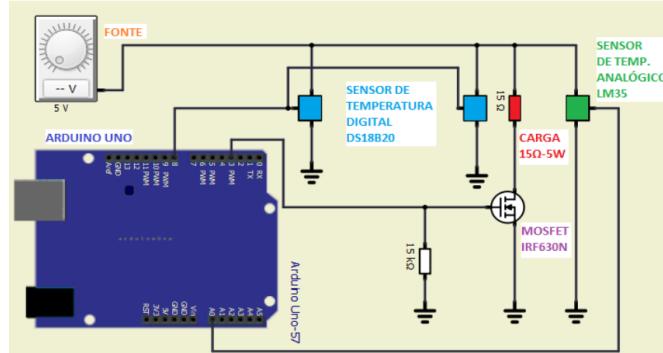
// Cria a fila para enviar os valores do ADC
adcQueue = xQueueCreate(10, sizeof(uint16_t));
// Sintaxe para a criação das tarefas
// função, nome da task, tamanho, parâmetros de entrada, prioridade, identificador
xTaskCreate(vMovingAverage, "Moving Average", 128, NULL, 1, NULL);
xTaskCreate(vReadADC, "ReadADC", 128, NULL, 1, NULL);

```

essas duas tarefas trabalham de forma coordenada, em que a tarefa “vReadADC” lê os valores do ADC e os envia para a fila “adcQueue”. Em paralelo, a tarefa “vMovingAverage” aguarda os valores na fila, realiza o cálculo da média móvel com base nos últimos 10 valores lidos e, posteriormente, envia o resultado para o computador para o monitoramento.

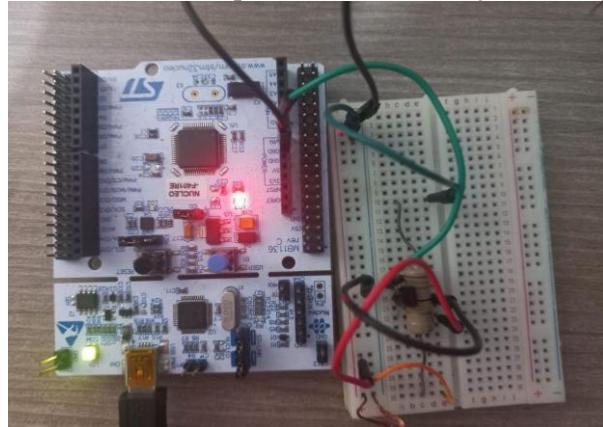
Por fim, foi montado um circuito eletrônico (Figura 2) baseado em BASTOS (2021) para testes preliminares de funcionamento do controlador e do medidor de temperatura. Modificações foram realizadas para adaptar o circuito à plataforma NUCLEO-F401RE, como a mudança dos pinos de saída utilizados, a utilização exclusiva do sensor de temperatura analógico LM35 e, para montar a curva de temperatura, inicialmente o resistor foi conectado diretamente a fonte de 5V, sem a utilização do MOSFET (figura 3).

Figura 2: Esquema eletrônico para controle e medição de temperatura em um resistor de potência.



Fonte: BASTOS (2021).

Figura 3: Montagem do circuito eletrônico para controle e medição de temperatura em um resistor.



Fonte: autor.

A medição de temperatura foi realizada utilizando o sensor analógico LM35, conectado a uma entrada analógica da placa NUCLEO-F401RE. A leitura da tensão gerada pelo sensor, que é proporcional à temperatura, é convertida em valores digitais pelo ADC. Esses valores são processados por uma tarefa específica que realiza a leitura periodicamente e envia os dados para uma fila (adcQueue), onde ficam armazenadas as amostras mais recentes.

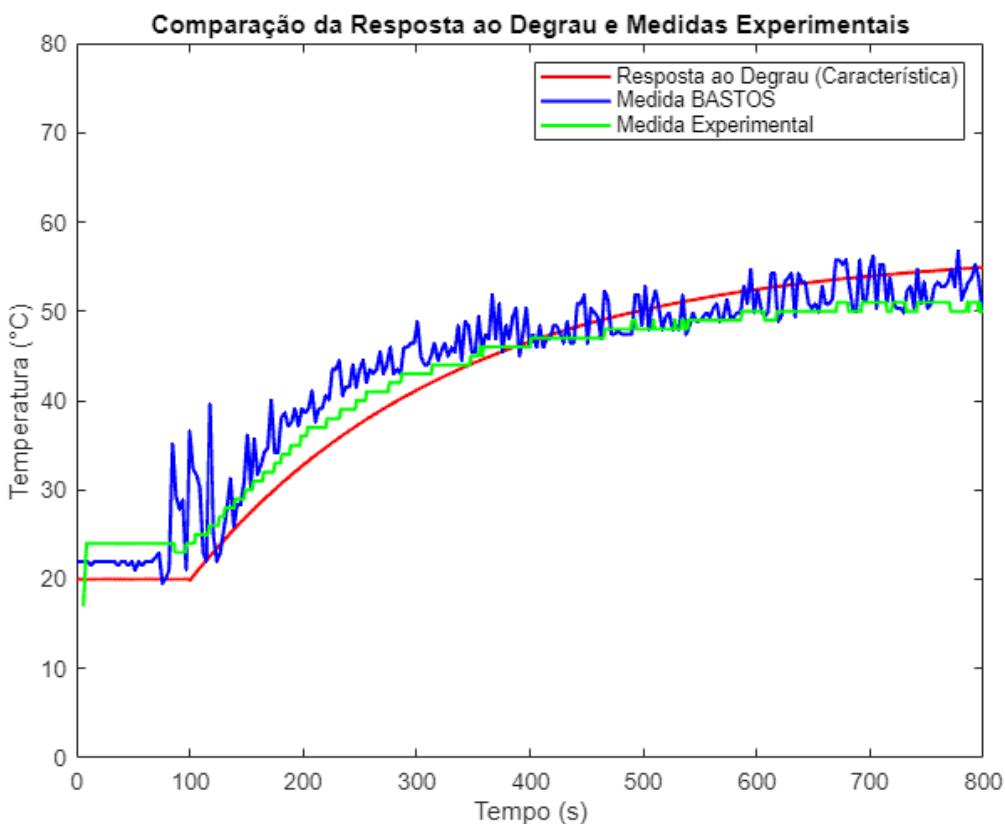
Para suavizar as variações das leituras e obter uma estimativa mais estável da temperatura, foi implementado, no programa, um filtro de média móvel. Esse filtro soma as últimas dez leituras do ADC e calcula a média, reduzindo ruídos e flutuações nos dados. Após esse processamento, a

temperatura filtrada é enviada ao computador por meio de uma comunicação serial, permitindo a visualização das leituras em tempo real em um terminal. Dessa forma, o sistema realiza a medição da temperatura de forma precisa e estável, além de garantir uma comunicação eficiente com o computador para monitoramento contínuo.

## RESULTADOS E DISCUSSÃO

Para manter o foco do trabalho na aplicação de controle de temperatura, o circuito de aquecimento e monitoramento foi baseado no trabalho de Bastos (2021), que realiza a medição de temperatura de cinco resistores de valores diferentes. Para efeito de testes, foi considerado apenas o resistor de 15R-2W neste trabalho. A Figura 4 mostra a resposta ao degrau característica do resistor utilizado, comparado aos resultados de Bastos (2021) e deste trabalho, onde é possível observar uma semelhança significativa entre os resultados. Ao se analisar o comportamento da temperatura ao longo do tempo, fica evidente um aumento gradual da temperatura até que o sistema atinja um equilíbrio térmico. Para a resposta ao degrau, o valor foi inicialmente ajustado para 20°C.

Figura 4: Comparação das medidas.



Fonte: autor.

A comparação entre os resultados foi feita por meio do cálculo das diferenças médias percentuais entre as medições experimentais e a curva característica. Os resultados obtidos mostram que a diferença média percentual entre a montagem de Bastos e a curva característica foi de 35,19%, enquanto que a diferença média percentual entre a medição experimental da montagem atual e a curva característica foi de 24,51%. Isso indica que a medição experimental atual apresenta uma menor discrepância em relação à curva característica do que a montagem de Bastos (2021), reforçando a consistência dos dados obtidos neste trabalho. Os testes iniciais foram realizados com o objetivo de traçar a curva de comportamento térmico do resistor utilizando um RTOS embarcado em um microcontrolador ARM, sem ainda implementar, de fato, um sistema de controle de temperatura. Os resultados apresentados aqui são preliminares e parciais, representando apenas uma etapa inicial do projeto. Na sequência, será implementado um sistema de controle para ajustar automaticamente a temperatura de acordo com as necessidades do sistema.

## **CONCLUSÕES**

Os resultados iniciais deste trabalho indicam que a aplicação de um RTOS na medição de temperatura apresenta semelhanças significativas com os resultados obtidos em estudos anteriores, mesmo utilizando um paradigma de programação distinto. Comparando os gráficos obtidos durante os testes preliminares com os do trabalho de Bastos (2021), observou-se que, apesar das diferenças nas abordagens e ferramentas utilizadas, o comportamento térmico do sistema foi semelhante, sugerindo a eficácia do método implementado com o *FreeRTOS*.

Os resultados apresentados são preliminares e se concentram na análise do comportamento térmico do resistor sem a aplicação de um sistema de controle automatizado. A curva de resposta obtida servirá como base para o desenvolvimento de um controlador mais sofisticado, que permitirá ajustes dinâmicos e precisos da temperatura.

Os próximos passos deste projeto incluem a implementação de um sistema de controle automatizado, utilizando o *FreeRTOS* para gerenciar as tarefas relacionadas ao controle térmico de forma eficiente. Espera-se que, com a adição deste controlador, os resultados finais sejam ainda mais alinhados com os objetivos inicialmente propostos.

Dessa forma, este trabalho contribui para uma compreensão mais aprofundada dos benefícios do uso do RTOS em sistemas embarcados, abrindo caminho para futuras otimizações e novos projetos na área de controle de processos.

## **CONTRIBUIÇÕES DOS AUTORES**

R.R. foi responsável pela conceitualização do projeto, pela administração do trabalho e pela revisão e edição do resumo. A.A.L.M. e R.R. contribuíram com a curadoria e análise dos dados, além de estarem envolvidos na pesquisa, metodologia e design da apresentação dos dados. R.R e A.J.A disponibilizaram as ferramentas necessárias para a realização do trabalho. A.A.L.M., L.F.C.R.M. e S.B.S. foram responsáveis pelo desenvolvimento, implementação e teste do *software*, além de redigir a versão original do resumo. Todos os autores participaram da revisão final do trabalho e aprovaram a versão submetida.

## **AGRADECIMENTOS**

Agradeço à minha família pelo apoio e aos meus colegas que, com suas contribuições e companheirismo, enriqueceram essa jornada. Agradeço também a todos os autores que colaboraram neste trabalho. Em especial, meu sincero agradecimento ao professor R.R, cuja orientação e dedicação foram fundamentais em cada etapa deste projeto, tornando possível sua realização e meu crescimento acadêmico.

## **REFERÊNCIAS**

AMAZON WEB SERVICES. **FreeRTOS**: Real-time operating system for microcontrollers and small microprocessors. EUA:AWS, 2024. Disponível em:<<https://www.freertos.org>>. Acesso em: 07 ago. 2024.

BASTOS, C. A.; RECH, R. Estudo de métodos de controle para um sistema térmico. In: **12º Congresso de Inovação, Ciência e Tecnologia do IFSP (CONICT)**. São Paulo: Instituto Federal de São Paulo (IFSP), 2021. p. 1–5. Disponível em:  
<<https://ocs.ifsp.edu.br/conict/xiiconict/paper/view/7375/2538>>. Acesso em: 07 ago. 2024.

KOPETZ, H.; STEINER, W. **Real-Time Systems**: Design principles for distributed embedded applications. 3. ed. Boston: Springer Cham, 2022. 406 p.  
<https://doi.org/10.1007/978-3-031-11992-7>.

MORAIS, J. **RTOS**: Um ambiente multi-tarefas para sistemas embarcados. Brasil: Embarcados, 2018. Disponível em: <<https://embarcados.com.br/rtos-sistema-operacional-de-tempo-real/>>. Acesso em: 17 jul. 2024.

SOUZA, F. **O que são sistemas embarcados?** Brasil: Embarcados, 2022. Disponível em: <<https://embarcados.com.br/o-que-sao-sistemas-embarcados/>>. Acesso em: 17 jul. 2024.