

ESTUDO DO SISTEMA OPERACIONAL DE TEMPO REAL FREERTOS APLICADO A MICROCONTROLADORES

Gabriel Lezan Nitz¹, Ana Teruko Yokomizo Watanabe².

¹ Acadêmico do Curso de Engenharia Elétrica – CCT

² Orientadora, Departamento de Engenharia Elétrica – CCT – ana.watanabe@udesc.br

Os sistemas embarcados estão presentes nas mais diversas aplicações vigentes, exercendo grande influência no cotidiano das pessoas. Um dos principais elementos constituintes desses sistemas é o *software* embarcado, que traz consigo a lógica que rege os comportamentos e interpretações dos periféricos de entrada e saída. Nessa temática, surgem os sistemas operacionais, responsáveis por introduzir uma camada de abstração entre *hardware* e código, além de gerenciar os recursos disponíveis no sistema (NOEGAARD, 2005).

Um sistema operacional de tempo real (RTOS) é um programa que gerencia os recursos e a execução de tarefas de maneira temporal. Apresenta algumas vantagens em aplicações com sistemas embarcados, como: confiabilidade, escalonabilidade, previsibilidade, rápida performance e portabilidade entre plataformas. Assim, um RTOS é capaz de controlar a execução de tarefas de acordo com o tempo previsto para estas, de forma que a característica de previsibilidade é vinculada ao sistema pela necessidade de todos os prazos estipulados serem cumpridos. O sistema é composto por um núcleo (*kernel*) que é responsável por escalonar os algoritmos e gerenciar os recursos (LI, 2003).

Existem vários RTOSs disponíveis no mercado, e alguns dos critérios para a escolha do sistema ideal para o seu projeto são: popularidade, custo, portabilidade, escalabilidade e ferramentas disponíveis. Para este trabalho foi empregado o sistema FreeRTOS, que é distribuído com a licença MIT. Esse sistema é um *kernel* que organiza as aplicações em coleções de tarefas independentes para execução, com base na prioridade vinculada a cada uma. A prioridade de cada tarefa é definida via código no momento da sua criação. É compatível com cerca de 20 compiladores e mais de 30 arquiteturas de processadores diferentes, podendo ser considerado uma biblioteca que confere a capacidade de multitarefas (BARRY, 2016). O FreeRTOS oferece várias funcionalidades para serem utilizadas em um projeto de tempo real, das quais podemos citar: tarefas, filas, semáforos, *timers* de *software* e grupos de eventos.

Para o estudo, o sistema operacional foi embarcado em uma placa Arduino UNO, que dispõe do microcontrolador ATmega328p. O estudo se deu com o projeto de 9 aplicações empregando diferentes funcionalidades do FreeRTOS, sendo desenvolvidas 5 no ambiente de desenvolvimento integrado (IDE) do Arduino e 4 no *software* Microchip Studio.

Um dos projetos desenvolvidos foi o controle de uma granja avícola utilizando o FreeRTOS no Arduino por meio do Microchip Studio. Dessa forma, a temperatura da granja é lida uma vez a cada segundo por meio de um *timer* de *software* com um sensor, que neste caso é substituído por um potenciômetro para controle da simulação. Um LED verde pisca a cada leitura da temperatura. Com isso, deseja-se que a temperatura se mantenha dentro de uma faixa preestabelecida, então se o ambiente estiver quente é habilitada uma tarefa que liga um ventilador, e se estiver frio é habilitada uma tarefa que liga uma lâmpada incandescente. Além disso, é possível ativar uma buzina que indica que a porta da granja pode ser aberta. Para isso é

criado um grupo de eventos com dois eventos, que devem ocorrer para ativar a buzina, conforme são indicados a seguir:

- Evento 1: ocorre se a temperatura estiver dentro da faixa preestabelecida. Um LED amarelo é aceso para indicar que a temperatura está nesse intervalo.
- Evento 2: ocorre se um botão for pressionado, indicando assim a abertura da porta por meio do acionamento da buzina. A leitura do botão só é permitida caso a temperatura esteja na faixa preestabelecida.

As tarefas são criadas com prioridades iguais. A Figura 1 apresenta o circuito com os componentes utilizados no projeto. Um relé tem como função acionar a lâmpada incandescente e o outro acionar o ventilador, representado pelo motor.

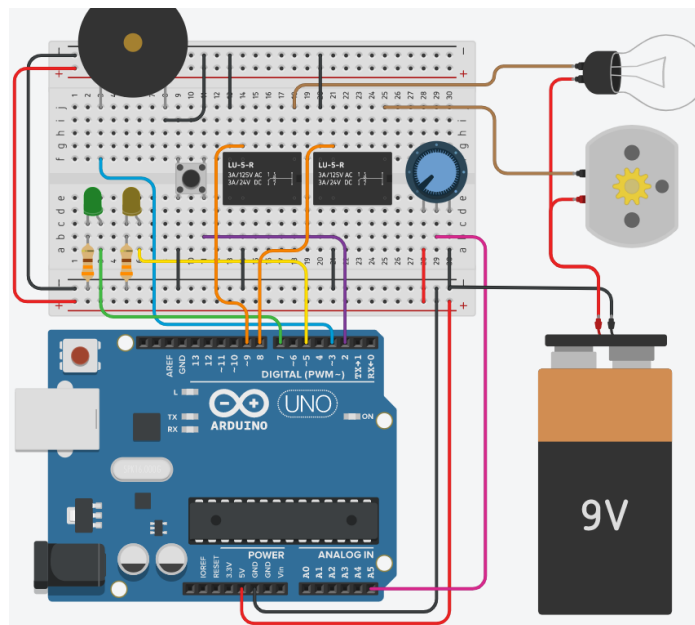


Figura 1. Projeto de controle de uma granja avícola

Como resultado do estudo obteve-se o entendimento do sistema operacional de tempo real FreeRTOS e de sua vasta aplicabilidade em sistemas que possuem prazos para execução, o que permitiu o desenvolvimento de uma apostila sobre o assunto para utilização na disciplina de Aplicações Avançadas em Microprocessadores do curso de Engenharia Elétrica do Centro de Ciências Tecnológicas da Udesc. Para a continuidade do estudo, é almejada a utilização do microcontrolador ESP32, que oferece maior poder de processamento que o Arduino UNO.

Palavras-chave: Sistema de tempo real. Microcontroladores. FreeRTOS.

Referências

NOEGAARD T. *Embedded Systems Architecture - A comprehensive Guide for Engineers and Programmers*. Elsevier, 2005.

LI Q. *Real Time concepts for Embedded Systems*. CMPBooks, 2003.

BARRY Richard. *Mastering the FreeRTOS Real Time Kernel: a hands-on tutorial guide*. 2016.