

Avaliação de Desempenho dos Sistemas Operacionais Embarcados MQX Lite e FreeRTOS aplicados a Robótica Móvel

Fernando Emilio Puntel¹, Anderson Luiz Fernandes Perez²

¹Laboratório de Computação Espacial
Centro Regional Sul
Instituto Nacional de Pesquisas Espaciais
Santa Maria, RS, Brasil

²Programa de Pós-Graduação em Tecnologias da Informação e Comunicação
Laboratório de Automação e Robótica Móvel
Universidade Federal de Santa Catarina
Araranguá, SC, Brasil

fernandopuntel@gmail.com, anderson.perez@ufsc.br

Resumo. *O avanço na robótica faz com que um robô desempenhe cada vez mais funções, onde em muitos casos é necessário a presença de um sistema operacional embarcado, capaz de gerenciar os recursos computacionais. Por isso, a escolha de um sistema operacional embarcado deve ser criteriosa. Neste trabalho, são descritos os resultados obtidos na comparação de desempenho entre os sistemas operacionais embarcados MQX Lite e FreeRTOS em um robô.*

1. Introdução

Robôs estão cada vez mais presentes em nosso dia a dia, se tornando indispensáveis em algumas tarefas. Com o avanço das tecnologias embarcadas, os componentes que fazem parte de um robô estão cada vez menores e com maior poder de processamento. Com todos esses avanços, a robótica que antes era somente braços manipuladores, aplicados à construção de algum bem de consumo, também passou a ser utilizada no auxílio do ser humano.

Robôs também estão presente na indústria e nas pesquisas, realizando tarefas perigosas, repetitivas e estressantes, que antes eram realizadas pelo ser humano. Com isso, cada vez mais possuem uma elevada complexidade no hardware, além da necessidade de realizar tarefas em prazos específicos. Este tipo de robô necessita que o software saiba manipular o hardware da melhor maneira possível, justificando o uso de um SOE (Sistema Operacional Embarcado).

Neste trabalho será descrito a avaliação dos sistemas operacionais embarcados FreeRTOS e MQX Lite em termos de eficiência no gerenciamento e na execução das tarefas de um robô móvel explorador.

Além desta introdução, este trabalho está dividido em mais 4 seções. A seção 2 apresenta definições de sistemas operacionais embarcados, dando ênfase nos dois SOEs utilizados neste trabalho. A seção 3 apresenta conceitos básicos de robótica móvel e

descreve o robô explorador, o qual foi utilizado para realizar os experimentos com os dois SOEs. Os resultados com MQX Lite e FreeRTOS são apresentados na seção 4. Na seção 5 é mostrado as considerações finais a respeito dos experimentos com ambos SOEs.

2. Sistemas Operacionais Embarcados

Geralmente aplicações embarcadas não necessitam de mecanismos complexos de proteção de memória e podem ser construídos em plataformas que não provém uma MMU (*Memory Management Unit*), os microcontroladores que podem ser baseados em arquiteturas RISC (*Reduced Instruction Set Computer*) ou CISC (*Complex Instruction Set Computer*), trocando eficiência de pipeline por densidade de código [Marcondes et al. 2006]. Um SOE é utilizado quando for necessário realizar uma atividade complexa, tendo assim que dividir o problema em tarefas específicas. Desta forma, o o programa é dividido em várias tarefas ou threads. Para este trabalho foram escolhidos dois SOEs afim de avaliar o desempenho de ambos:

- **MQX Lite** é baseado no sistema operacional embarcado MQX porém para plataformas com recurso limitados. O MQX Lite possui uma quantidade menor de componentes comparado ao MQX, mantendo as funcionalidades básicas de um SOE, permitindo assim que tarefas que consomem menos de 4 KB (Kilobyte) de RAM (*Random Access Memory*), sejam executados. A grande vantagem deste sistema é que foi totalmente projetado para plataforma escolhida neste trabalho (Freedom Board KL25Z);
- **FreeRTOS** é um sistema operacional embarcado de código aberto. Seu núcleo é totalmente escrito em C, e possui 4 (quatro) arquivos fontes, que são facilmente portáveis e de fácil manutenção, seu escalonador é baseado em prioridades fixas e suporta um elevado número de tarefas [El Amine and Mohamed 2013]. Atualmente o FreeRTOS é um dos SOE mais utilizados em aplicação, isso deve-se ao SOE ser de código aberto e por possuir uma comunidade de usuários ativas.

3. Robô Explorador

Um robô é classificado como móvel quando pode se locomover no ar, terra ou na água. Um robô móvel possui diferentes tipos de sensores e em sua maioria possui mais que uma função, como por exemplo, um robô que aspira o chão, este é preciso ter percepção de local, saber monitorar a quantidade de bateria disponível, entre outros. Neste trabalho foi utilizado um robô explorador móvel, onde o robô precisa interagir com o ambiente a todo tempo e realizar tarefas dentro de prazos específicos, como por exemplo, desviar de um objeto reconhecido por sensores.

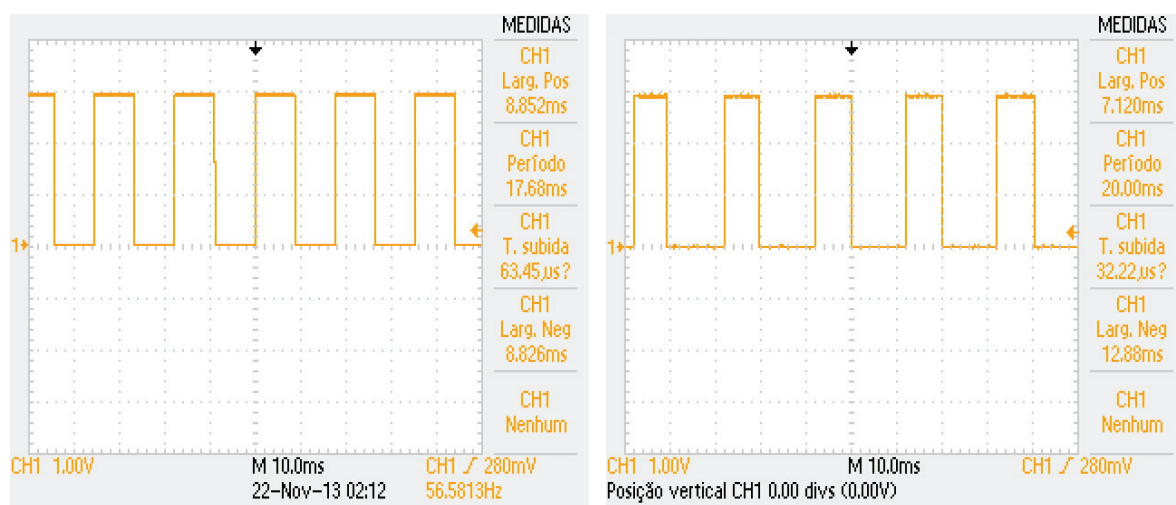
O robô explorado de ambientes [Perez et al. 2013] utilizado nos experimentos é composto por 6 sensores ultrassom (3 na dianteira do robô e 3 na traseira), responsáveis pela identificação de objetos próximos ao robô e dois motores que funcionam como atuadores, para que o robô possa se movimentar no ambiente. A base mecânica do robô é uma caminhonete de controle remoto, onde estão instalados todos os sensores, atuadores e o microcontrolador. O microcontrolador utilizado nestes experimentos é o KL25Z da Freescale [Freescale 2012], por ser uma plataforma de baixo custo, com suporte a SOE e de fácil aprendizado.

4. Avaliação dos Sistemas Operacionais Embarcados

Foram realizados 2 experimentos afim de avaliar os dois SOEs, MQX Lite e FreeRTOS. Os experimentos realizados foram: tempo relacionado a troca de contexto (Subseção 4.1), e tempo gasto na execução de uma seção crítica (Subseção 4.2). Os gráficos apresentados em cada experimento foram obtidos com o auxílio de um osciloscópio.

4.1 Experimento 1 - Tempo relacionado a troca de contexto

Para que todas as tarefas entrem em execução é preciso que o escalonador de tarefas realize a troca de contexto, ou seja, selecionar a próxima tarefa que irá ocupar o processador. Neste experimento são avaliados a troca de contexto dos SOEs em questão. A Figura 1 (a) ilustra o tempo de troca de contexto do MQX Lite e a Figura 1 (b) ilustram a troca de contexto no FreeRTOS.



(a) Figura 1 - Troca de contexto no MQX Lite

(b) Figura 1 - Troca de contexto no FreeRTOS

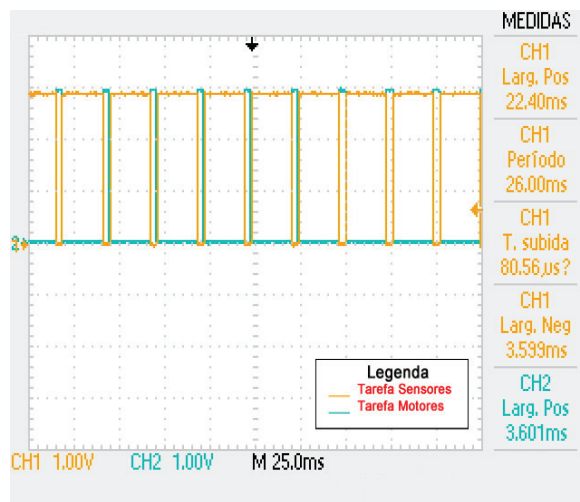
Após concluído o experimento, constatou-se que o FreeRTOS realizou a troca de contexto em 7.120 milissegundos (ms) e teve um desempenho superior ao MQX Lite que realizou a troca de contexto em 8.852 ms. Ambos realizaram a troca de contexto em tempo hábil, não danificando a parte física do robô.

4.2 Experimento 2 - Tempo gasto na execução de uma seção crítica

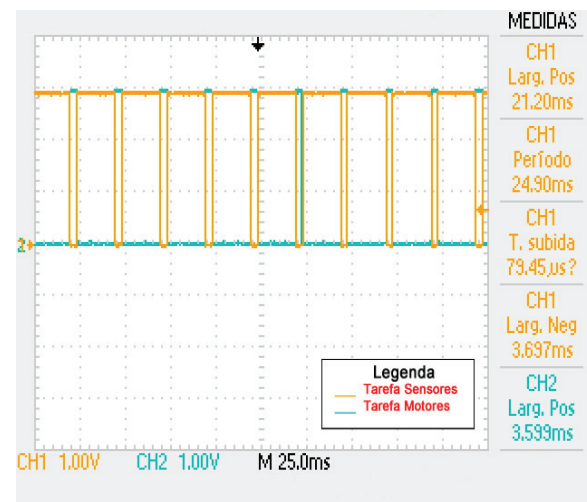
Neste experimento foi definida uma seção crítica com três variáveis do tipo inteiro, estas variáveis representam a leitura dos três sensores dianteiros. O escalonador utilizado foi o FIFO (*First In First Out*) em ambos os SOE, nestes experimentos as duas tarefas possuem a mesma prioridade.

Para este experimento a tarefa “sensores” realiza a leitura dos dados dos sensores, acessa a seção crítica somente para alterar o valor das variáveis, após isso libera a seção crítica, depois a tarefa “atuadores” solicita o acesso a seção crítica, somente para leitura dos dados e libera a seção em seguida. A Figura 2 (c) ilustra o tempo gasto na execução da seção crítica utilizando o MQX Lite e a Figura 2 (d) ilustram no SOE FreeRTOS.

Percebe-se que o MQX Lite demorou em média 22.40 ms para executar a tarefa de leitura de dados dos sensores e 3.6 ms para executar a tarefa dos atuadores, enquanto o FreeRTOS demorou em média 21.20 ms na tarefa sensores e 3.6 ms na tarefa motores.



(c) Figura 2 - Tempo gasto na seção crítica utilizando MQX Lite



(d) Figura 2 - Tempo gasto na seção crítica utilizando FreeRTOS

5. Conclusões

Afim de comprovar e avaliar os benefícios do uso de um SOE, este trabalho apresentou uma avaliação dos SOEs MQX Lite e FreeRTOS em aplicações de robótica móvel. Para a avaliação foram realizados alguns experimentos com ambos os sistemas operacionais.

Foram avaliados os tempos de troca de contexto e acesso a seção crítica de ambos os SOEs em um robô explorador de ambientes, dotado de 2 motores e 6 sensores ultrassônicos. Com os resultados dos experimentos constatou-se que ambos os sistemas realizaram operações em tempo hábil sem danificar o robô. Fazendo assim com que o robô conseguisse se adaptar as mudanças do ambiente.

Por fim, conclui-se que em algumas situações o uso de SOE se faz necessário, devido a facilidade para gerenciamento de recursos e controle sobre o hardware.

References

- El Amine, C. M. and Mohamed, O. (2013). A localization and an identification system of personnel in areas at risk using a wireless sensor network. In *Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2013 International Conference on*, pages 127–131. IEEE.
- Freescale (2012). *FRDM - KL25Z User's Manual*. Freescale, 1 edition.
- Marcondes, H., Junior, A. S. H., Wanner, L. F., Cancian, R. L., Santos, D. M., and Fröhlich, A. A. M. (2006). Epos: Um sistema operacional portátil para sistemas profundamente embarcados. In *Workshop de Sistemas Operacionais*.
- Perez, A. L. F., Puntel, F. E., Nandi, G. C. S., and Schueroff, J. (2013). Desenvolvimento de um robô explorador para ambientes indoor. *Revista Técnico Científica do IFSC*, 1(5):650.