**APLICABILIDADE DE SISTEMAS OPERACIONAIS DE TEMPO REAL (RTOS) PARA AUTOMAÇÃO RESIDENCIAL**

Rodrigo Fontan Arruda Cedrim[[1]](#footnote-1)

Edson Struminski[[2]](#footnote-2)

Isaac Nunes de Oliveira[[3]](#footnote-3)

Agnaldo Cardozo Filho[[4]](#footnote-4)

**RESUMO**

Os sistemas operacionais em tempo real (RTOS) existem há muito tempo e tem como principal função garantir que os prazos para cumprimento das tarefas estabelecidos no *software* sejam atendidos, utilizando diversos recursos para possibilitar que mesmo um número grande de tarefas possam ser atendidas no prazo correto. No desenvolvimento de sistemas de automação residencial, a execução dessas tarefas dentro do prazo não é crítica, porém é extremamente desejável que haja uma previsibilidade do sistema. Esta pesquisa analisa a necessidade de utilizar um RTOS num dispositivo embarcado para um sistema de automação residencial, considerando os principais recursos dos RTOS e um sistema de automação residencial fictício.

**Palavras-chave:** RTOS. Automação Residencial. Sistema Embarcado.

**1 INTRODUÇÃO**

O desenvolvimento da automação residencial está em constante evolução, seja por meio da ampliação da gama de itens automatizados ou pela melhoria dos sistemas existentes. Essas mudanças ocorrem de forma rápida e constante, sempre objetivando o aperfeiçoamento do sistema como um todo, buscando mais integração e facilitando a vida das pessoas.

Nos últimos anos o setor de automação teve um crescimento elevado, partindo de simples portões eletrônicos a sistemas complexos. Segundo Bolzani (2004, p. 45),

A automação é capaz de melhorar o estilo de vida dos moradores aumentando conforto, segurança e eficiência energética, pois seus sistemas englobam iluminação, entretenimento, segurança, telecomunicações, climatização e sistemas de áudio e vídeo.

Para se desenvolver um sistema que englobe a automação de tantos itens é necessária a utilização de vários sensores, atuadores, circuitos eletrônicos e ao menos uma Interface Homem-Máquina (IHM). Tantas tarefas, com diferentes prioridades, necessidades, respostas, precisando ser atendidas nas suas respectivas *deadlines* (tempo máximo para execução da tarefa), geram dificuldade de se trabalhar com o *Superloop*, especialmente por conta do controle de tempo e da modularidade do *software*.

A complexidade desses sistemas de automação e a necessidade de atender as *deadlines* de cada uma das tarefas, levantou a possibilidade de se utilizar um Sistema Operacional de Tempo Real (RTOS) para melhorar a resposta do sistema. Segundo Borges (2011), este tipo de sistema constitui uma ferramenta poderosa para gerenciar a complexidade, facilitar o reuso e aumentar a portabilidade do software, reduzindo assim o *time-to-market*. Esse tipo de software modular é de grande ajuda para sistemas de automação residencial, em que cada residência é diferente e precisa de soluções específicas.

O objetivo de estudo do presente trabalho é analisar a necessidade de um RTOS num sistema de automação residencial de médio porte, que englobe os principais itens de conforto, segurança e eficiência energética.

**2 AUTOMAÇÃO RESIDENCIAL**

De acordo com Wortmeyer, Freitas e Cardoso (2005), a automação residencial representa o emprego de tecnologias ao ambiente doméstico, com o objetivo de propiciar conforto, praticidade, produtividade, economia, eficiência e rentabilidade. Com o avanço da automação, tais tecnologias passaram a ser controladas remotamente, seja limitado dentro da residência (WiFi, RF, Bluetooth) ou até mesmo de qualquer lugar do mundo por meio da Internet.

Existem diversos elementos necessários para a obtenção de informações, transporte de dados, processamento e execução da ação necessária. Para um sistema de automação de médio porte, são necessários controladores, sensores, atuadores, barramentos e interfaces, descritos a seguir.

O controlador é o elemento central do sistema de automação, responsável pelo processamento de todas as informações. Segundo Sgarbi (2007), todos os elementos do sistema se conectam ao controlador, enviando e recebendo informações. É nele que se encontra o *software* programado e onde todas as decisões serão tomadas, acionando atuadores, atualizando a IHM, desligando as luzes dos ambientes, ajustando a temperatura, entre outros.

De acordo com Accardi e Dodonov (2012), sensores são dispositivos que medem e monitoram grandezas físicas e eventos (temperatura, umidade, presença etc.), convertendo-as em um valor passível de manipulação por sistemas computacionais. Eles se conectam com o controlador e seguem enviando as informações na frequência determinada na sua construção ou conforme solicitado pelo controlador. É utilizando a informação dos sensores que o controlador decide quando acionar os atuadores.

Os atuadores são dispositivos eletromecânicos que fazem as movimentações do sistema. Eles recebem os comandos do controlador e ativam os equipamentos automatizados, sejam eles portas, janelas, persianas, fechaduras magnéticas, sirenes, indicadores luminosos, etc. (ACCARDI; DODONOV, 2012).

O barramento ou rede domótica é por onde trafegam todos os dados do sistema de automação. É o cabeamento por onde passam os dados dos sensores e os comandos dos atuadores. Segundo Sgarbi (2007, p. 46),

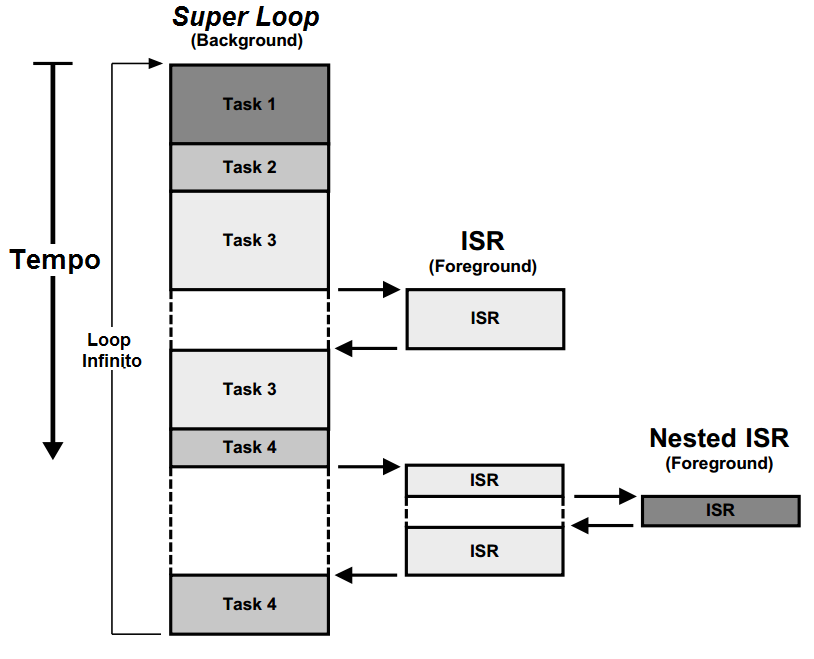
As redes se baseiam em aplicações, onde uma rede separada e independente é utilizada para cada função, portanto podem existir redes destinadas a segurança, à detecção de incêndios, ao controle de acessos, à climatização, à informática, entre outros.

As interfaces por sua vez são responsáveis pela interação com os habitantes da residência, mostrando as informações, permitindo o controle. Tais interfaces podem ser controles com *displays*, um *smartphone* ou qualquer dispositivo que acesse a internet, permitindo que todos da casa operem o sistema (BOLZANI, 2004).

**3 SISTEMA OPERACIONAL EM TEMPO REAL (RTOS)**

Com a evolução dos sistemas embarcados, veio o aumento na quantidade e complexidade dos componentes integrados ao controlador, fazendo com que a estrutura chamada comumente de *superloop*se torne ineficiente, já que se perde o controle da taxa de atualização desses componentes, fazendo com que o sistema não seja confiável (LABROSSE, 2011). A Figura 1 mostra o funcionamento de um sistema *superloop*.

**Figura 1** – *Software* mostrando as tarefas em *Background* e em *Foreground*



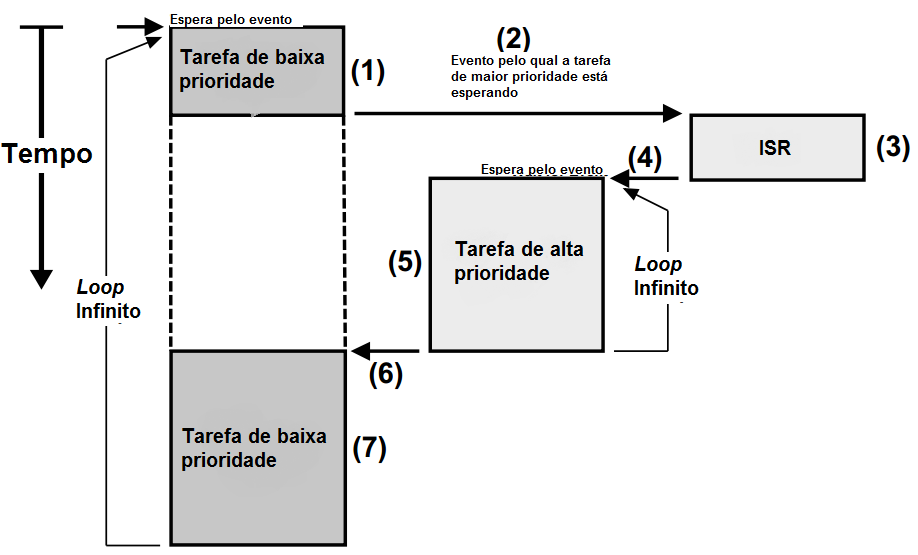
**Fonte:** Adaptado de Labrosse (2011).

Segundo Borges (2011), sistemas de tempo real são sistema capazes de garantir que determinada tarefa será executada sempre dentro do limite pré-determinado (*deadline*). Uma definição bem aceita de sistema de tempo real foi feita por Laplante (2004, p. 5), “um sistema de tempo real é aquele que deve satisfazer explicitamente restrições de tempo de resposta, podendo ter consequências de risco ou falhas se não satisfeitas suas restrições”.

Após verificar as definições dos autores acima, é possível concluir que o objetivo de um sistema operacional de tempo real é previsibilidade de quando cada uma das tarefas será concluída. É um erro comum acreditar que um sistema operacional de tempo real é mais rápido que o normal, sendo ele apenas mais preciso no que diz respeito ao tempo.

O processo de utilização de um RTOS requer que seu trabalho seja dividido em pequenas tarefas, cada uma responsável por uma parte do trabalho. Cada tarefa acredita ter a unidade central de processamento (CPU) para si e executa a tarefa num determinado tempo. A implementação de tarefa geralmente ocorre na forma de *loop* infinito (FIGURA 2).

**Figura 2** – Linha de execução de um RTOS



**Fonte:** Adaptado de Labrosse (2011).

Os sistemas operacionais de tempo real se dividem basicamente em três categorias, de acordo com as consequências das falhas, são elas: *Hard Real Time*: consiste de um sistema de tempo real onde sua falha pode acarretar consequências desastrosas, tanto financeiramente quanto a perda de vidas humanas (LABROSSE, 2011); *Soft Real Time*: sistema em que uma falha ou perda de prazos não causa dano, mas o sistema deixa de cumprir sua função (LABROSSE, 2011); *Firm Real Time*: é o meio termo entre os dois de cima, falhas acumuladas podem transformar um *Soft Real Time* em um *Hard Real Time* (AROCA, 2008).

Para o sistema de automação residencial simulado vamos analisar um *Firm Real Time*, pois para um sistema de automação residencial que inclua itens como segurança e eficiência energética, existe um claro risco no caso do sistema parar de funcionar por qualquer razão.

**4 ANÁLISE DA AUTOMAÇÃO RESIDENCIAL COM O RTOS**

**Ao c**onsiderar uma residência de porte médio e que tenha os seguintes itens automatizados: Segurança: sensores de presença e de quebra de vidro nas esquadrias; Multimídia: tela do projetor, som e iluminação; Atuadores: janelas, cortinas, portões, fechaduras magnéticas etc.; Eficiência Energética: controle de iluminação, detectando presença e desligando iluminação durante o dia/em cômodos vazios, controle da água do jardim etc.; Que conte com ao menos uma IHM na residência mais a quantidade desejada controlada externamente via internet.

**Figura 3** – Sistema de automação residencial centralizado

**Fonte:** Elaborado pelos Autores.

Com um sistema centralizado, onde temos apenas um controlador e todas as informações necessárias chegam e partem dele por meio dos barramentos construídos na residência, é possível perceber a complexidade que o *software* pode chegar. Fazendo uma análise mais profunda do sistema acima, vamos determinar alguns requisitos para que o sistema opere com qualidade: Sensores de presença e quebra de vidro devem ser verificados a cada 100ms; A resposta do sistema multimídia não pode ser superior a 150ms; Os atuadores devem ser acionados em no máximo 200ms após solicitados; O sistema de eficiência energética deve verificar a presença em todos os cômodos da casa e a luminosidade externa a cada 5 minutos, e acionar os aspersores de água a cada 8 horas durante 20 minutos; Atualização da IHM interna a cada 100ms e das externas a cada 150ms.

Os prazos acima foram escolhidos para garantir que o sistema modelo opere com agilidade e sem sobrecarregar o controlador. Vale destacar que existem *n* outras combinações que também funcionariam a contento, ficando a cargo do desenvolvedor estabelecer as metas e prioridades do sistema desenvolvido.

De acordo com Farines, Fraga e Oliveira (2000), o que mais importa nos sistemas de tempo real é a previsibilidade, que possibilita saber como o sistema vai se comportar mesmo sob falhas pontuais. Tal previsibilidade dá ao desenvolvedor a possibilidade de integrar ao seu sistema recursos para se recuperar e prevenir falhas localizadas, reiniciando alguma tarefa, limpando determinada variável etc.

Ao analisar o funcionando do sistema de automação residencial escolhido, tanto o *superloop* como o RTOS conseguem operar de maneira satisfatória, porém o primeiro não consegue cumprir todas as *deadlines* citadas acima, especialmente se o controlador escolhido for de pequeno porte. Para utilizar satisfatoriamente os recursos presentes num RTOS, Borges (2011) defende que o controlador deve possuir ao menos 4 Kbytes de memória RAM, se considerarmos *deadlines* superiores a 1ms, sendo assim, existem controladores que têm o mínimo para operar um RTOS e cumprir os prazos (quando superiores a 1ms) e não conseguem cumprir os mesmos prazos se operando no modo *superloop*. A razão disso não está relacionada à velocidade de processamento, mas sim à imprevisibilidade desse modo.

Quanto à aplicação de tais tecnologias, o RTOS exige conhecimentos mais específicos, fazendo com que apenas 28% dos sistemas embarcados o utilizem (BORGES, 2011). Apesar do *superloop*ser mais simples, desenvolvedores experientes não devem sentir dificuldade em trabalhar com RTOS, por conta de sua boa documentação.

No que tange a modularidade do *software*, que seria, por exemplo, aplicar o produto numa outra residência, com um número diferente elementos automatizados (janelas, cômodos, IHMs), o RTOS traz clara vantagem, já que uma das suas principais características é a facilidade de modulação do código, fazendo com que o seu *software* seja mais adaptável, aumentando a possibilidade de reuso. No caso do *superloop,* mesmo desenvolvendo o código com essa necessidade em mente, adaptar o *software* para outra residência se torna um trabalho complicado, especialmente para ajustar cada função a sua respectiva *deadline.*

A questão da modularidade está ligada diretamente ao *time-to-market*, pois quanto mais adaptável for o *software* desenvolvido, mais rapidamente seu projeto é adaptado as diferentes residências. De acordo com pesquisa efetuada por Borges (2011), onde foi traçado um perfil do desenvolvimento de sistemas embarcados no Brasil, os maiores desafios da área são *time-to-market* e o custo, e que o cronograma inicial para novos projetos é geralmente mais de 6 meses, sendo prorrogado por, em média, 3 meses. Os prazos acima são para produtos novos, a adaptação de uma residência para outra tem de acontecer em prazos muito mais curtos.

A comparação entre os dois métodos de trabalho em relação ao custo deve considerar vários fatores, como no sistema fictício analisado foi determinado as *deadlines* de cada uma das tarefas, esse vai ser o ponto chave. Um controlador, utilizando *superloop* e devidamente ajustado para atender os prazos acima tem custo inferior a um controlador que consiga operar um RTOS com todos os seus recursos, porém há ainda a possibilidade de utilizar um pseudo-sistema de tempo real, que teria apenas parte das funções de um RTOS, possibilitando o uso de um controlador com custo reduzido.

Ao concluir a análise da aplicabilidade de um RTOS num sistema de automação residencial, temos que a utilização de um RTOS é um diferencial ao lançar um produto desse tipo no mercado, pois a modularização do código permite que o desenvolvedor adeque seu produto as mais diferentes residências num curto prazo e garanta que todas as tarefas sejam realizadas no tempo prometido. Porém, existem custos extras na utilização destes sistemas, como contratação de mão de obra especializada e a possível compra de um RTOS comercial (existem RTOS gratuitos, mas que não são tão bem documentados). Tais custos podem ser revertidos em longo prazo, se aproveitando da modularização e reuso do *software* para melhor aproveitar o *time-to-market.*

**5 CONCLUSÃO**

A utilização de RTOS nos sistemas de automação residencial tem suas vantagens e desvantagens bem definidas, ficando sob responsabilidade do desenvolvedor analisar o que é mais importante no projeto desenvolvido.

Conforme apontado por Borges (2011), no Brasil apenas 28% dos sistemas embarcados utilizam RTOS, se aproveitando da modularidade e do reuso do código facilitado pelo sistema. Para automação residencial, tais características são fundamentais e justificam o uso de um RTOS para essa aplicação, já que comparando com o *Superloop*, a diferença no custo do desenvolvimento do produto final de suas possíveis alterações não é elevado.

**REFERÊNCIAS**

ACCARDI, A.; DODONOV, E. Automação residencial:elementos básicos, arquiteturas, setores, aplicações e protocolos. **T.I.S. – Tecnologias, Infraestrutura e Software***,* São Carlos, v.1, n.2, p.156-166, nov. 2012.

AROCA, R.V. **Análise de sistemas operacionais de tempo real para aplicações de robótica e automação***.* 2008. 154f. Dissertação (Mestrado em Engenharia Mecânica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.

BOLZANI, C. A. M. **Desenvolvimento de um simulador de controle de dispositivos residenciais inteligentes:** uma introdução aos sistemas domóticos. 2004. 130 f. Dissertação (Mestrado em Engenharia Elétrica) – Departamento de Engenharia Elétrica, Escola Politécnica da Universidade de São Paulo, São Paulo, 2004.

BORGES, R. W. **Aplicabilidade de sistemas operacionais de tempo real (RTOS) para sistemas embarcados de baixo custo e pequeno porte***.* 2011. 152 f. Dissertação (Mestrado em Engenharia Elétrica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2011.

FARINES, J.; FRAGA, J.; OLIVEIRA, R. **Sistemas de tempo real**. Florianópolis, 2000. 208p.

LABROSSE, J. J. *µC/OS-III*: **The real-time kernel**. Florida: Micriµm Press, 2011. 934p.

LAPLANTE, P.A.; OVASKA S.J. **Real-time systems design and analysis:** tools for the Practicioner. 4.ed. Hoboken: John Wileyand Sons, 2012. 560p.

SGARBI, J.A. **Domótica inteligente:**Automação Residencial Baseada em Comportamento*.* 2007. 107 f. Dissertação (Mestrado em Engenharia Elétrica) – Centro Universitário da FEI, São Bernardo do Campo-SP, 2007.

WORTMEYER, C.; FREITAS, F.; CARDOSO, L. **Automação residencial:** Busca de tecnologiasvisando o conforto, a economia, a praticidade e a segurança do usuário. In: II Simpósio de Excelência em Gestão e Tecnologia SEGeT2005. [S.I.: s.n.], 2005.

**Data do recebimento:** 22 de dezembro de 2016

**Data de avaliação:** 09 de janeiro de 2017

**Data de aceite:** 05 de fevereiro de 2017

1. Discente do Curso de Engenharia Mecatrônica do Centro Universitário Tiradentes – UNIT. E-mail: rodrigocedrim@gmail.com [↑](#footnote-ref-1)
2. Docente do Curso de Engenharia Mecatrônica do Centro Universitário Tiradentes – UNIT. E-mail: duboisedson@gmail.com [↑](#footnote-ref-2)
3. Docente do Curso de Engenharia Mecatrônica do Centro Universitário Tiradentes – UNIT. E-mail: isaacnunesdeoliveira@gmail.com [↑](#footnote-ref-3)
4. Docente do Curso de Engenharia Mecatrônica do Centro Universitário Tiradentes – UNIT. E-mail: eng.agnaldofilho@gmail.com [↑](#footnote-ref-4)