Sistema tolerante a falhas para monitoramento de temperatura com redundância em hardware

Sara Guimaraes Negreiros[1], Leila Maria de Freitas Sousa [1], Verônica Maria Lima Silva[1]

[1]Universidade Federal Rural do Semi-Árido (UFERSA) – Pau dos Ferros – RN – Brasil

sguimaraaes@gmail.com, leilafreitas159@gmail.com, veronica.lima@ufersa.edu.br

Abstract. *In several environments, temperature is a critical influence factor and needs to be monitored continuously to ensure that the system performs the correct action in relation to the local temperature value. In this work a fault-tolerant system with the Atmega 328 microcontroller is proposed, which implements hardware redundancy to deal with temperature sensors.*

Resumo. *Em diversos ambientes a temperatura é um fator crítico de influência e precisa ser monitorada de modo contínuo para garantir que o sistema realize a ação correta diante do valor da temperatura local. Neste trabalho é proposto um sistema tolerante a falhas com o microcontrolador Atmega 328 que implementa redundância em hardware para lidar com sensores de temperatura.*

1. Introdução

Atualmente é cada vez mais comum a utilização de sistemas de computação que auxiliam nas mais diversas atividades. Ao passo que a tecnologia evolui e se torna mais popular, mais pessoas passam a utilizar e depender fortemente do desempenho desses sistemas. Consequentemente, vê-se a necessidade de tratar os problemas que podem vir a afetar os sistemas.

2. Tolerância a falhas

A tolerância a falhas é um mecanismo cujo objetivo consistem em alcançar a dependabilidade, que indica a qualidade do serviço fornecido por um dado sistema, bem como a confiança depositada no serviço por ele oferecido. Segundo Weber (2003), os principais atributos de dependabilidade são:

- Confiabilidade: é a capacidade de atender a especificações, dentro de condições definidas, durante um período de funcionado determinado;
- Disponibilidade: é o atributo mais usado em sistemas de missão crítica e consistem na probabilidade de o sistema estar em operação num dado instante de tempo;
- Segurança de funcionamento (*safety*): probabilidade de o sistema descontinuar suas funções sem provocar danos a outros sistemas ou pessoas que dele dependam;

- Segurança (*securiy*): remete a proteção contra falhas, visando a privacidade, autenticidade, integridade e não-repudiabilidade dos dados;
- Mantenabilidade: facilidade de realizar a manutenção do sistema, o que engloba a probabilidade de restauração do sistema dentro de um período determinado;
- Testabilidade: capacidade de realizar testes em atributos internos ao sistema. Nessa perspectiva, quanto maior a testabilidade, melhor a mantenabilidade.
- Comprometimento do desempenho (*perfomability*): relacionado à queda de desempenho provocado pela ocorrência de falhas (sistema continua a operar, contudo, degradado em desempenho).

3. Redundância de hardware

A aplicação de redundância para implementação de técnicas de tolerância a falha pode aparecer na forma de redundância de informação, redundância temporal, redundância de hardware e redundância de software. Todas estas formas têm algum impacto no sistema e demandam custo e desempenho. O uso da redundância pode ser feito tanto para a detecção de falhas quanto para os mascaramentos delas.

A redundância de hardware consiste na replicação de componentes físicos e pode se apresentar de forma passiva ou ativa. Na primeira os elementos redundantes são utilizados para fazer o mascaramento de falas. Assim, todos os elementos executam a mesa tarefa e o resultado é determinado por votação.

Redundância modular tripla (TMR) é uma das técnicas mais populares e consistem em mascarar falhas em um componente triplicando ele e votando entre as saídas. A votação pode ocorrer por maioria (2 em 1) ou através do valor médio. Para tornar o sistema mais confiável pode-se construir o votador com componentes de alta confiabilidade.

Na redundância de hardware ativa (ou dinâmica), a tolerância a falhas é estruturada pelo uso de técnicas para detecção, localização e recuperação, não havendo o mascaramento de falhas. Esse tipo de redundância tem grande uso em aplicações que suportam permanecer em estados errôneos por períodos curtos de tempo, intervalo de tempo este em que é feita a detecção do erro e a recuperação para um estado isento de falhas. A técnica *Standby* Sparing é um exemplo de redundância ativa, e consistem na utilização de módulos redundantes para fazer a substituição do componente em uso no caso em que este esteja falho.

A implementação da *Standby* Sparing pode ser feita por duas abordagens. Na primeira (*Hot Standby Sparing*) os módulos redundantes estão em funcionamento sincronizado com o módulo de operação, o que afere um maior consumo de energia. Já na segunda (*Cold Standby Sparing*), as réplicas encontram-se desligadas até que exista a necessidade de uma substituição. A inicialização das réplicas confere um tempo adicional à execução do sistema.

4. Implementação do sistema para monitoramento da temperatura

Nesta esquematização os sensores são então ativados ou desativados a partir de falhas ou novos pareamentos. A principal motivação em utilizar os transistores é impedir

que os sensores fiquem sempre acionados de modo contínuo, consumindo energia e diminuindo sua vida útil. O acionamento é realizado enviando um pulso para a base dos transistores que estão conectadas aos pinos digitais do arduino. Neste circuito ainda são utilizados diodos 1N4003 e resistores para garantir o funcionamento ideal e seguro dos demais componentes. Dessa forma, caso um sensor seja desativado por falha ou não precise estar em funcionamento em todo o tempo, a chave é fechada e há menor consumo de energia por parte do circuito. É então possível perceber que a implementação descrita segue o padrão da redundância de hardware ativa *Cold Standby Sparing*.

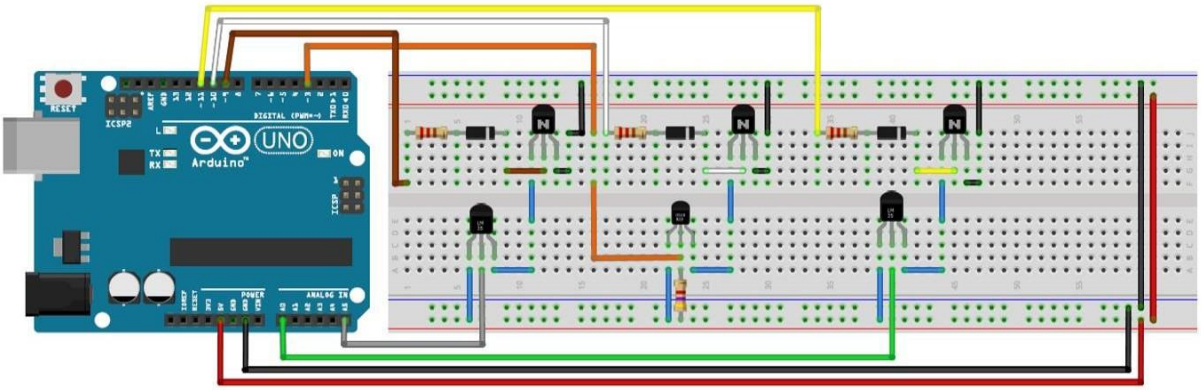

Figura 1. Esquema do Circuito

Pelo fluxograma da Figura 2 observa-se que inicialmente é realizada a medição do Sensor i cujo valor é destinado para um bloco de Display que também pode ser entendido como a atuação final da medida de temperatura. Após 5 segundos a medida do Sensor i é comparada com um Sensor ii e desde que esteja dentro de uma variação de ±15% do valor medido, o sistema não detecta nenhuma variação alta de temperatura e continua seu funcionamento normalmente. Caso contrário, as medidas desses dois sensores são destinadas para um bloco que consiste em comparar esses valores com um Sensor iii, de maior prioridade. Neste bloco o Sensor iii é definido como o sensor juiz, pois com base no valor dele o sistema irá verificar qual dos outros sensores possui maior diferença para desativar esse sensor. Por fim, o sistema verifica se existem sensores disponíveis para atualizar as listas i e ii, se não houver, o pino 13 do arduino é ativado como um sinal de alerta para o usuário, caso contrário, as listas são atualizadas e o sistema continua em operação. O algoritmo que implementa essa lógica no microcontrolador é disponibilizado no repositório *guimaraaes/sistemaToleranteAFalhasHardware* do GitHub.

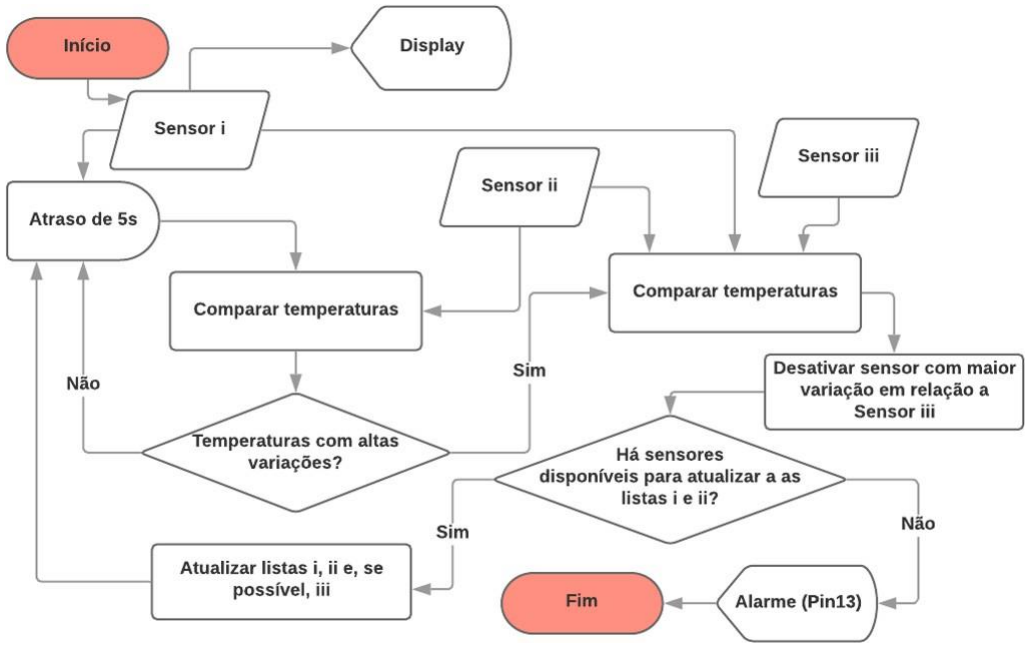

Figura 2. Fluxograma do funcionamento do sistema de tolerância a falhas

5. Considerações finais

Desse modo, o sistema mostrou-se eficaz para monitorar as variações de temperatura entre os sensores e eliminar os dados de sensores que apresentavam probabilidade de estarem danificados. Apesar de terem sido utilizados apenas três sensores, é possível aumentar essa quantidade apenas replicando as chaves analógicas do circuito e as estruturas condicionais do código.

Referências

Redundância modular dupla - Dual modular redundancy. In: Wikipédia: a enciclopédia livre. Disponível em: <https://pt.qwertyu.wiki/wiki/Dual_modular_redundancy>. Acesso em: 09 dez. 2019.

WEBER, T. S. Tolerância a falhas: conceitos e exemplos. Intech Brasil. São Paulo. v. 52, p. 32-42, 2003. Disponível em: <http://www.inf.ufrgs.br/~taisy/disciplinas/textos/Dependabilidade.pdf>. Acesso em: 06 dez. 2019.