

Um Protótipo de Sistema para Auxílio a Deficientes Visuais em Semáforos de Trânsito

Daniel Souza dos Santos¹, Leonardo Venancio Correia²,
Luis Claudio de Oliveira Silva³

¹Curso de Ciência e Tecnologia – Universidade Federal do Maranhão (UFMA)
São Luís – MA – Brasil

²Curso de Engenharia da Computação – Universidade Federal do Maranhão (UFMA)
São Luís – MA – Brasil

³Coordenação do Curso de Engenharia da Computação
Universidade Federal do Maranhão (UFMA) São Luís – MA – Brasil

danielss.eng@gmail.com, leonardo.venancio@aluno.ecp.ufma.br

lcoliveira.silva@gmail.com

Abstract. *Taking into consideration the difficulties that people with visual or hearing impairment face on a daily basis, a method is presented to help them cross lanes which have traffic lights. Using a real-time operational system for embedded systems – FreeRTOS, on an Arduino UNO R3 with visual and audible components, a prototype system was produced which can realize multiple simultaneous tasks without losing their synchronicity, offering greater reliability and safety for users.*

Resumo. *Levando em consideração dificuldades que o público portador de deficiência visual ou auditiva enfrenta diariamente, é mostrado nesse artigo um método para auxílio dos mesmos na travessia de vias com semáforos de trânsito. Com a utilização de um sistema operacional de tempo real gratuito para sistemas embarcados – FreeRTOS, em um Arduino UNO R3 com componentes luminosos e sonoros, produziu-se um protótipo de sistema que consegue realizar múltiplas tarefas simultâneas sem perder a sincronia, oferecendo maior confiabilidade e segurança para os usuários.*

1. INTRODUÇÃO

Segundo dados obtidos no Censo Demográfico 2010 à luz das recomendações do Grupo de Washington em 2018, 3,4% da população brasileira têm grande dificuldade permanente ou não consegue enxergar, enquanto que 1,1% se enquadram nos mesmos critério para a deficiência auditiva. [IBGE 2018]

Neste artigo é proposto um protótipo de dispositivo de baixo custo, fácil instalação, fácil uso e confiável para, assim, haver uma implementação em semáforos para auxílio a deficientes, tanto visuais quanto auditivos, em momentos de travessia de vias de trânsito, em concordância com a resolução do Conselho Nacional de Trânsito – CONTRAN N° 704 de 10/10/2017 [CONTRAN 2017].

De acordo com [Oliveira 2018], os sistemas de tempo real são sistemas que têm a capacidade de atender requisitos temporais explícitos. O sistema do estudo pode ser

considerado como tendo o requisito temporal com *deadline* brando, ou seja, a perda do *deadline* de uma tarefa não tem consequências catastróficas, porém seu benefício decresce com o seu atraso, não o perdendo completamente. Entretanto, o sistema dispõe de uma plataforma embarcada possuindo apenas um núcleo de processamento e conta com a necessidade da criação de múltiplas tarefas principais e que precisam estar sincronizadas.

Para suprir a necessidade temporal, o dispositivo dispõe da utilização de um kernel sistema operacional de tempo real gratuito e de código aberto para atender requisitos de sistemas de tempo real de *deadline* rígido em sistemas embarcados, o FreeRTOS [Turci 2017]. A biblioteca do FreeRTOS se torna uma opção viável por conta da pouca necessidade de memória, alta confiabilidade e rápida execução, conjuntamente com seu escalonador, este que é preemptivo e cooperativo, além de possuir tarefas assíncronas e estas serem invocadas pelo o escalonador. [Goyette 2007]

Como o principal objetivo dessa aplicação é auxiliar os portadores de deficiências visuais na travessia de vias com semáforos, sem deixar de também favorecer os portadores de deficiência auditiva, ressalta-se que o sistema proposto não consegue resolver o problema do estudo por completo e sim que poderá trazer mais autonomia e segurança para tais usuários.

2. MATERIAIS E MÉTODOS

O sistema por completo é composto por um *hardware*, o qual é descrito na subseção 2.1 referente aos materiais, e um *software*, que será o tema de estudo na subseção 2.2, referente aos métodos.

2.1. MATERIAIS

Para a confecção do sistema embarcado foram usados os materiais dispostos na tabela 1, bem como as suas respectivas quantidades e finalidades.

Tabela 1. Itens usados para construir o protótipo

Item	Qtde	Propósito
Botão	1	Botão de sinalização de intenção de travessia
LED de cor verde	2	Simbolizam Faróis Verdes (Via e Pedestre)
LED de cor vermelha	2	Simbolizam Faróis Vermelhos (Via e Pedestre)
LED de cor amarela	1	Simboliza Farol Amarelo (Via)
mini-LED de cor azul	1	Sinalização Visual de Funcionamento
Resistor de 220 Ω	6	Limitação de corrente de cada LED
Resistor de 1000 Ω	1	Limitação de corrente da conexão do Botão
Display TM1637	1	Exibição do tempo de abertura restante dos semáforos
Arduino Uno R3	1	Processamento de dados, entrada e saída
Buzzer	1	Simboliza Sirene de emissão de sinais sonoros
Breadboard	1	Agrupamento das conexões dos componentes

O projeto criado com o software *Fritzing* mostra em detalhes como componentes são conectados em suas devidas portas, o qual foi usado para a elaboração de protótipo e pode ser visto na figura 1.

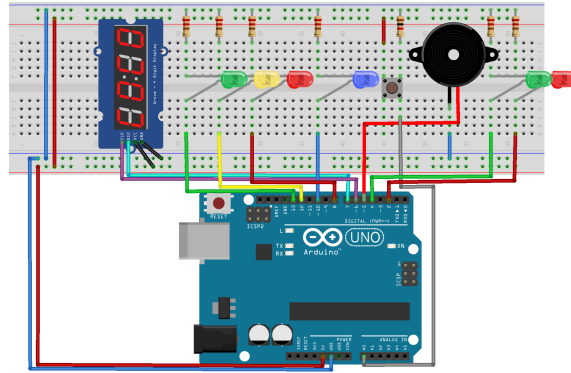


Figura 1. Esquema de montagem com Arduino

Vale salientar que o protótipo é designado para apenas uma única faixa de pedestres e conta apenas com um botão, porém estes podem ser expandidos facilmente aumentando a quantidade dos componentes usados e ajustando o software. Ressalva-se que apenas um display TM1637 foi usado para ser dividido e mostrar ambos os tempos de abertura restantes em cada semáforo em seus quatro dígitos, sendo os dois dígitos mais à esquerda usados para a via principal e os outros dois para mostrar o tempo do pedestre.

2.2. MÉTODOS

Para melhor visualização do funcionamento do sistema, uma Rede de Petri – RP com peso igual a um foi criada para mostrar, de forma inequívoca, como cada um dos estados se comporta em cada um dos ciclos. A Rede de Petri pode ser vista na Figura 2:

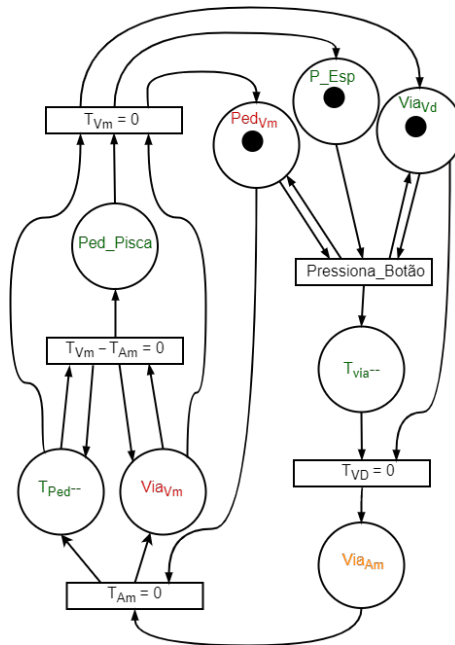


Figura 2. Rede de Petri do Semáforo

De acordo com [Cardoso and Valette 1997], em uma RP, existem quatro elementos principais: posições (círculos sem preenchimento), marcações (círculos pequenos preenchidos), transições (retângulos) e arcos direcionados indicando origem e destino

das marcações. Em funcionamento, uma transição retira simultaneamente a marcação de cada posição origem, quando sua condição é atingida, para inserir marcações nas posições destino de acordo com a quantidade de arcos. Cada posição fica ativa enquanto tiver marcações.

O Algoritmo foi desenvolvido na IDE do Arduino usando as bibliotecas do FreeRTOS e do display TM1637. O sistema consiste em um semáforo comum com 3 faróis, um semáforo de pedestre com 2 faróis, um sistema sonoro e um sistema de interrupção.

A aplicação foi dividida em quatro principais tarefas: a primeira para gerenciar o acender e apagar dos LEDs, a segunda para gerenciar o número que é mostrado no display, a terceira para controlar a emissão dos sinais sonoros e a quarta para verificar de tempos em tempos se o botão foi pressionado. Geralmente para sincronização de eventos e interrupção de rotinas de serviços, semáforos binários são usados para bloquear as tarefas quando uma interrupção ocorre.

Para o desenvolvimento do dispositivo seguiu-se os padrões do Conselho Nacional de Trânsito, o CONTRAN, os quais indicaram momento, intermitência, duração e frequência dos sinais sonoros do dispositivo, entre outros.

No estado inicial da aplicação, o sistema deve estar com o farol verde para a via, vermelho para o pedestre, e o sistema fica em espera, representados pelas marcações nas posições da RP por P_Esp , Ped_{Vm} e Via_{Vd} , que representam, respectivamente, o sistema do pedestre em espera, o sinal vermelho para o pedestre e o sinal verde na via.

Ao mesmo tempo, enquanto não houver interrupção por pressionar o botão de travessia, representada pela transição $Pressiona_Botão$, o dispositivo deve emitir um som intermitente com as seguintes características:

- Intermitência: 0,5 Hz (1 ciclo a cada 2s)
- Duração: 60 ms ($\pm 2ms$)
- Frequência: 950 Hz ($\pm 10Hz$)

Ocorre uma interrupção assim que o usuário pressiona o botão implementado na aplicação, então o LED azul acende ao mesmo tempo que se ativa um aviso sonoro de curta duração para alertar o usuário que o sistema entrou em funcionamento, a marcação em P_Esp é consumida e uma marcação é adicionada em T_{Via--} , que representa o contador decrescendo uma unidade por unidade de tempo sendo exibido no display. As marcações em Ped_{Vm} e Via_{Vd} são consumidas e repostas logo após a transição, mantendo os semáforos verde e vermelho acesos.

Quando o tempo deste contador regressivo chega a zero, representado pela transição $T_{Vd} = 0$ para o mudança de estado do semáforo da via, consome as marcações em T_{Via--} e Via_{Vd} e insere uma em Via_{Am} , fazendo com que o semáforo verde da via apague e o amarelo da via acenda.

Quando o tempo do sinal amarelo chega a zero, na transição $T_{Am} = 0$, as marcações em Via_{Am} e Ped_{Vm} são consumidas e inseridas em T_{Ped--} e Via_{Am} , fazendo com que o sinal amarelo da via e o sinal vermelho do pedestre se apaguem, e o sinal vermelho da via e o verde do pedestre se acendam. Simultaneamente, inicia o parâmetro para o sinal sonoro de início do tempo de travessia (silvo inicial do tempo de verde do foco do pedestre).

- Intermitência: 1 pulso único, antecedendo o sinal sonoro de travessia.
- Duração: 160 ms ($\pm 5ms$)
- Frequência: 2000 Hz ($\pm 10Hz$) decrescendo gradativamente até 500 Hz ($\pm 10Hz$)

Logo em seguida, é emitido o sinal sonoro de travessia (tempo de verde do foco do pedestre), que tem os seguintes parâmetros:

- Intermitência: 1 Hz (ciclo/s)
- Duração: 160 ms ($\pm 5ms$)
- Frequência Modulada: 2000 Hz ($\pm 10Hz$) + 500 Hz ($\pm 10Hz$)

Por último, quando o contador regressivo do pedestre se aproxima do final de sua execução, equivalente ao sinal amarelo, o sinal sonoro de advertência de encerramento de travessia (tempo de vermelho intermitente do foco de pedestre), denominado pela transição $T_{Vm} - T_{Am} = 0$, inserindo uma marcação em *Ped_Pisca*, fazendo com que o sinal verde do pedestre comece a piscar e os parâmetros sonoros se tornem:

- Intermitência: 2 Hz (ciclo/s)
- Duração: 160 ms ($\pm 5ms$)
- Frequência Modulada: 2000 Hz ($\pm 10Hz$) + 500 Hz ($\pm 10Hz$)

Logo após esse estado, quando contador geral chega ao seu limite, representado por $T_{Vm} = 0$, todas as marcações são consumidas e reinseridas nas marcações do estado inicial, fazendo com que além do sinal da via voltar para o verde e vermelho para o pedestre, o LED azul se apaga, pois este funciona basicamente para sinalizar o funcionamento do início da mudança de estados.

3. RESULTADOS

A Figura 3 mostra o protótipo do projeto já montado, com código carregado e em pleno funcionamento.

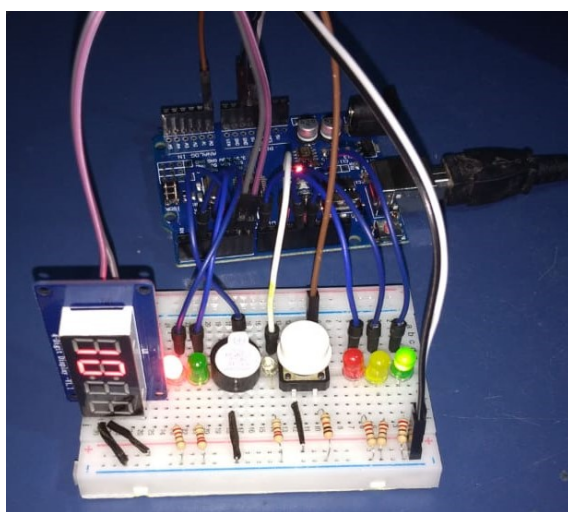


Figura 3. Protótipo Implementado

O sistema conseguiu cumprir o objetivo proposto, necessitando de alguns ajustes finos. A exemplo, a mudança de sinais sonoros simbolizando o início da travessia

ainda não ficou dentro dos padrões esperados e o número mostrado no display não está completamente sincronizado com a contagem.

Isso abre espaço para melhorias, tanto no algoritmo quanto no próprio protótipo. Porém, os sinais sonoros especificados no algoritmo podem ser bem entendidos para cada estado do sistema: sistema em espera; acionamento de passagem; sinal verde para o pedestre; aviso de advertência de fechamento do sinal; e fechamento do sinal.

Alguns outros detalhes não previstos pela resolução do CONTRAN foram adicionados ao algoritmo e ao sistema em si, como o som de frequência e duração distintas emitido pelo dispositivo quando o botão de travessia é pressionado e o sinal visual de sistema acionado, representado pelo LED azul, para alertar os usuários que não conseguem ouvir o sinal sonoro. Destacando outra implementação nova, esta é em relação ao estado de espera após pressionar o botão, este sistema esperará um tempo pré-determinado até poder emitir som novamente caso seja pressionado mais de uma vez.

Assim, como a utilização desse equipamento, é esperado que os usuários deste sistema se sintam representados e mais seguros ao atravessar estas vias e com esperança de diminuição do número de acidentes próximos a semáforos.

4. CONCLUSÃO

Como não existe somente a dependência de um sistema para mitigar ocorrências com deficientes na travessia das ruas (trata-se também de um problema social e educacional em relação aos motoristas e aos próprios pedestres), esta aplicação não eliminará todos os riscos.

Entretanto, diante deste projeto foi visto que um algoritmo com utilização da biblioteca FreeRTOS consegue explorar melhor o leque de possibilidades de um Arduino e, consequentemente, executar as tarefas propostas e com a eficiência esperada, já que não se trata de um sistema muito complexo. Contudo, foi essencial o uso de tal biblioteca. Já que se trata de um problema multitarefa e que o tempo de execução é crucial para o bom funcionamento da aplicação.

Como mostrado em [G1-MG 2017], essa aplicação já é utilizada em grandes cidades. Diante disso, é perceptível outro grande ganho com a utilização de um RTOS, a manutenção do sistema é mais fácil e eficiente, podendo ser encapsulada, pois está em nível mais alto. Comparado a um sistema utilizando apenas as funções *Setup* e *Loop*, como mostrado em [Pereira et al. 2018], seria necessário alterar o código inteiro e alterar a lógica, além de retrabalho para calcular possíveis atrasos na máquina de estados mostrada na Rede de Petri e sincronizar todas as tarefas.

Este trabalho abre espaço para estudos futuros, dos quais é possível destacar: a amplificação dos sons produzidos pelo equipamento e alimentação dos faróis; testes em um ambiente controlado para verificar sua confiabilidade; comparar esta proposta com as já existentes no mercado; aumentar a quantidade de vias controladas pelo processador central; entre outras incluindo desenvolver um produto final com estrutura protetora e acoplável a semáforos já existentes. Todo o código-fonte pode ser encontrado no *GitHub* nas contas dos autores do trabalho, bem como um vídeo demonstrativo referente ao seu funcionamento.

Referências

- AWS (2017). *The FreeRTOS™ Reference Manual: API Functions and Configuration Options*. Amazon Web Services.
- Barry, R. (2017). *Mastering the FreeRTOS™ Real Time Kernel: A Hands-On Tutorial Guide*. Real Time Engineers.
- Cardoso, J. and Valette, R. (1997). *Redes de petri*. Editora da UFSC.
- CONTRAN (2017). Resolução nº 703, de 10 de outubro de 2017. <http://pesquisa.in.gov.br/imprensa/jsp/visualiza/index.jsp?jornal=1&data=18/10/2017&pagina=184>. Acesso: 03/07/2019.
- G1-MG (2017). Mais um semáforo com dispositivo sonoro ajuda na travessia de deficientes visuais em belo horizonte. <https://g1.globo.com/minas-gerais/noticia/mais-um-semaforo-com-dispositivo-sonoro-ajuda-na-travessia-de-deficientes-visuais-em-belo-horizonte.ghtml>. Acesso: 03/07/2019.
- Giese, F. A., dos Santos, F. E., Almeida, H. F., Martins, M. P., de Campos Cardoso, L., and Pereira, M. H. R. (2016). Sistema para gerenciamento de semáforos para ambulâncias em situações emergenciais. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, 1:13.
- Goyette, R. (2007). An analysis and description of the inner workings of the freertos kernel. *SYSC5701: Operating System Methods for Real-Time Applications*.
- IBGE (2018). Censo demográfico 2010 – nota técnica 01/2018: Releitura dos dados de pessoas com deficiência no censo demográfico 2010 à luz das recomendações do grupo de washington. ftp://ftp.ibge.gov.br/Censos/Censo_Demografico_2010/metodologia/notas_tecnicas/nota_tecnica_2018_01_censo2010.pdf. Acesso: 22/07/2019.
- Lima, G. A., Mariano, J. G. O., Silva, S. A., and Lira, V. V. Desenvolvimento de um sistema microcontrolado para gerência de malha viária urbana. *Congresso Norte Nordeste de Pesquisa e Inovação*, page 7.
- Oliveira, R. S. (2018). *Fundamentos dos Sistemas de Tempo Real*. Amazon do Brasil, 1st edition. Paginação irregular. Livro digital.
- Pereira, V. H. M., Lima, W. C., and Amorim, L. E. (2018). Semáforo inteligente e acessibilidade pública. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, 6(3):155–188.
- Silva, D. M. (2016). Projeto educacional com base no sistema de automação de um semáforo para educação para o trânsito. http://www.waltenomartins.com.br/tcc_2016_daniel.pdf.
- Turci, L. O. (2017). Real-time operating system freertos application for fire alarm project in reduced scale. *International Journal of Computing and Digital Systems*, 6(4):197–204.