

# Sistema de Sinalização para Ciclistas

Karine Valença

*Engenharia de Software*

*Universidade de Brasília, FGA*

Gama, Brasil

valenca.karine@gmail.com

Wilton Rodrigues

*Engenharia de Software*

*Universidade de Brasília, FGA*

Gama, Brasil

wiltonsr94@gmail.com

**Abstract**—A fim de diminuir a quantidade de acidentes envolvendo ciclistas nas vias brasileiras foi proposto o objetivo de desenvolver um sistema de sinalização luminosa para permitir aos ciclistas indicarem suas intenções para os demais participantes do trânsito através de uma matriz luminosa. Utilizando o microcontrolador MSP430 da Texas Instruments foi possível atingir o objetivo proposto.

**Index Terms**—msp430, matriz de led, max7219, microcontrolador, ciclista, sinalização

## I. INTRODUÇÃO

### A. Revisão Bibliográfica

Notícias sobre acidentes envolvendo bicicletas são comuns no Brasil. Recentemente, em São Paulo, um ciclista morreu logo após ser atropelado e arrastado [1]. Dados de 2014, mostram que 1.357 ciclistas morreram vítimas de acidentes de trânsito no Brasil, além disso, em 2016, ocorreram 11.741 internações de ciclistas vítimas de acidentes [2]. De acordo com Departamento Nacional de Infraestrutura de Transportes (DNIT) [3] só no ano de 2011 foram 1.698 casos de acidentes envolvendo ciclistas. Sendo que 246, equivalente a 14.5%, acabaram na morte.

O site hg.org apresenta uma lista de dicas para evitar acidentes ao utilizar bicicleta. O site sugere aos ciclistas que eles se façam visíveis aos demais usuários das vias, e que utilizem sinais de mão para mostrar intenção de parar ou de mudar de faixa [4].

Existe uma série de sinais que podem ser utilizados pelos ciclistas para indicar suas intenções. O site mapmyrun [5], apresenta um lista com 10 sinais que podem ser utilizados a fim de evitar acidentes. Pode-se notar que, de fato, os sinais auxiliam a diminuir os acidentes de trânsito envolvendo ciclistas. Porém, alguns desses sinais não são tão intuitivos e podem não fazer sentido para os motoristas. Além disso, a grande quantidade de sinais pode gerar confusão até mesmo aos ciclistas.

### B. Justificativa

Pode-se notar que a visibilidade e sinalização por parte dos ciclistas é crucial para sua segurança no trânsito. Diante disso, este projeto tem como objetivo a criação de um sistema de sinalização eletrônico visando aumentar a segurança dos ciclistas. Espera-se que os usuários do sistema de sinalização eletrônico sofram menos acidentes causados por falta de visibilidade.

### C. Objetivos

O objetivo do projeto é desenvolver um sistema de sinalização, utilizando o MSP430, a fim de aumentar a visibilidade dos ciclistas durante seu trajeto para aumentar a segurança e confiança dos utilizadores deste meio de transporte.

### D. Requisitos

O sistema deve atender aos requisitos:

- Indicar sinal luminoso intermitente que fica ativo sempre que não houver outro sinal
- Indicar seta para a direita ou para a esquerda após clique do botão correspondente
- Indicar sobre parada quando o ciclista iniciar a freagem

O sistema não atende aos requisitos:

- Funcionar em dias chuvosos
- Possuir fonte de energia própria

### E. Benefícios

O sistema proporciona um equipamento de sinalização que ajuda os demais condutores a ter uma melhor visão dos ciclistas. Baseado nisto o principal benefício do sistema é a diminuição de ocorrências de acidentes envolvendo ciclistas.

## II. DESCRIÇÃO DO HARDWARE

### A. Lista de Materiais

Os materiais utilizados para a construção do Sistema de Sinalização para Ciclistas, foram:

- 1 MSP430 LaunchPad



Fig. 1. MSP430 LaunchPad. Fonte: <http://e2e.ti.com/>

- 1 Matriz de LED 8x8

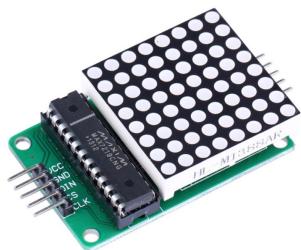


Fig. 2. Matriz de LED 8x8. Fonte: <http://www.huinfinito.com.br>

- 2 Protoboards

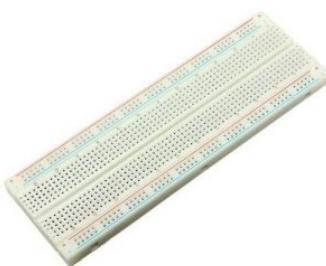


Fig. 3. Protoboard. Fonte: [www.filipeflop.com/](http://www.filipeflop.com/)

- Jumpers Macho-Macho e Macho-Fêmea

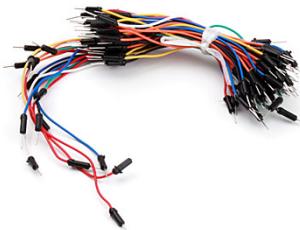


Fig. 4. Jumpers. Fonte: <http://www.msselettronica.com>

- 1 chave on-off-on

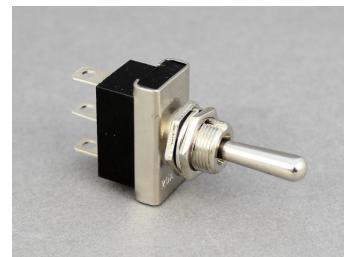


Fig. 5. Chave on-off-on. Fonte: <http://www.12voltplanet.co.uk/>

### B. Verificação dos componentes

Com o intuito de verificar se os componentes estavam funcionando conforme o esperado, forem feitos alguns teste simples de funcionamento.

*1) Verificação da Matriz de Led:* Para verificar se a matriz de led estava funcionando corretamente, foi ligada uma tensão de aproximadamente 3.3 volts no VCC e no DIN, e no GND, uma tensão de 0 volts.

**Esquemático:** A figura abaixo mostra o esquemático montado para o teste da matriz de led:

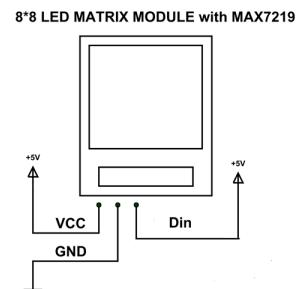


Fig. 6. Teste realizado com a matriz de led. Fonte: Autores

**Demonstração:** A figura abaixo mostra o teste realizado com a matriz de led:



Fig. 7. Teste realizado com a matriz de led. Fonte: Autores

2) *Verificação da Chave on-off-on:* Para verificar se a chave on-off-on estava funcionando corretamente, foi ligada uma tensão de aproximadamente 3.3 volts nos pinos 1 e 3 do botão. O pino 2 funcionou como saída e foi ligado em um resistor de 1000 ohm, que estava ligado em série a um Led.

**Demonstração:** A figura abaixo mostra o teste realizado com a chave on-off-on:

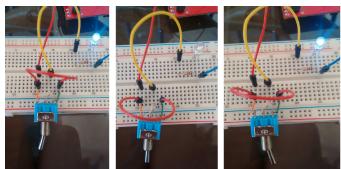


Fig. 8. Teste realizado com a chave on-off-on. Fonte: Autores

### C. Montagem

Para auxiliar no procedimento de montagem, foi feito um esquemático utilizando a ferramenta Fritzing<sup>1</sup>. A figura 9 apresenta o diagrama simplificado com os componentes de hardware que são utilizados no sistema.

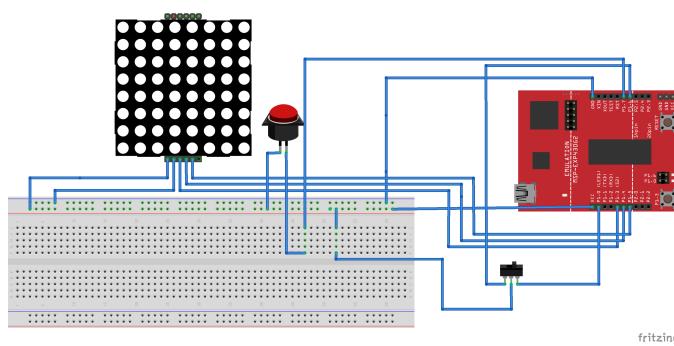


Fig. 9. Esquemático do hardware. Fonte: Autores

A matriz de led, que possui o multiplexador MAX7219, tem seu primeiro pino, VCC, ligado na entrada de energia de 3,3 Volts. O segundo pino, GND, é ligado no aterramento. O terceiro pino, DIN, que corresponde ao DATA IN, é ligado

no BIT3 do MSP. O quarto pino, CS, é ligado no BIT4 do MSP. E finalmente o quinto pino, CLK, que corresponde ao CLOCK, é ligado no BIT5 do MSP.

A chave on-off-on tem o pino central ligado no VCC. O pino da direita é ligado no BIT0 do MSP e o pino da esquerda é ligado no BIT6 do MSP.

A chave push-button tem um de seus pinos ligados no VCC, enquanto o outro pino fica ligado no BIT7 do MSP.

## III. DESCRIÇÃO DO SOFTWARE

### A. Diagrama UML

A figura 10 apresenta o diagrama UML do sistema. O sistema consiste de 3 arquivos, o arquivo main e 2 bibliotecas. O main realiza a chamada das funções das bibliotecas, realizando a execução do sistema em si. A biblioteca Desenho contém os desenhos que são exibidos pela matriz 8x8. A biblioteca MAX7219 contém todas as funções relacionadas à manipulação do multiplexador MAX7219.

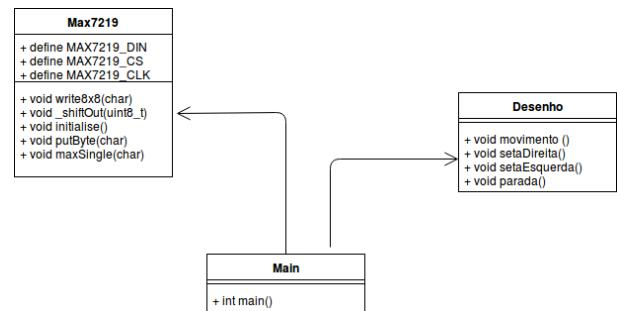


Fig. 10. Diagrama UML do sistema. Fonte: Autores

### B. Descrição do funcionamento

1) *Max7219.h:* Nesta biblioteca estão as funções de comunicação dos dados enviados pela placa MSP430, o multiplexador MAX7219 e a matriz de led.

```

void initialize(){
    Leve o pino CS para nível alto;
    Sete os pinos CS, DIN, CLK para saída;
}

void shift_out(entrada, clock, valor){
    Itere nos 8 bits da entrada;
    Se a entrada != 0
        Leve o valor da entrada para a matriz;
    Se não
        Limpe o valor da entrada da matriz;
    Sete o clock para nível alto;
    Sete o clock para nível baixo;
}

void put_byte(dado){
    Enquanto i > 0{
        Gere a máscara de ativação do led;
        Leve o CLK para o nível baixo;
        Se dado & máscara
  
```

<sup>1</sup><http://fritzing.org/home/>

```

    Leve o DIN para o nível alto;
Se não
    Leve o DIN para o nível baixo;
    Leve o CLK para o nível alto;
    Decremente o i;
}
}

```

```

void max_single(registrador , coluna){
    Leve o estado de CS para nível baixo;
    Execute put_byte(registrador);
    Execute put_byte(coluna);
    Leve o estado de CS para nível alto;
    Leve o estado de CS para nível baixo;
}

```

```

void write8x8(a, b, c, d, e, f, g, h){
    Execute mas_single(a);
    Execute mas_single(b);
    Execute mas_single(c);
    Execute mas_single(d);
    Execute mas_single(e);
    Execute mas_single(f);
    Execute mas_single(g);
}

```

2) *Desenho.h*: A biblioteca Desenho.h é a responsável pelo apoio na criação de funções para a exibição de imagens na matriz de led 8x8. Nesta biblioteca estão as funções setaDireita(), setaEsquerda(), movimentoContinuo() e parada(). O funcionamento básico de todas as funções desta biblioteca é o seguinte:

```

void setaEsquerda(){
    Executa write8x8(8 valores hexadecimais);
}

```

As figuras a seguir ilustram o funcionamento de cada uma das funções baseado no funcionamento descrito.

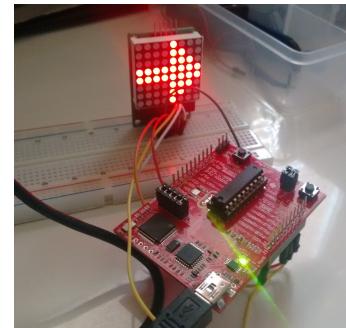


Fig. 12. Função setaDireita() em funcionamento. Fonte: Autores

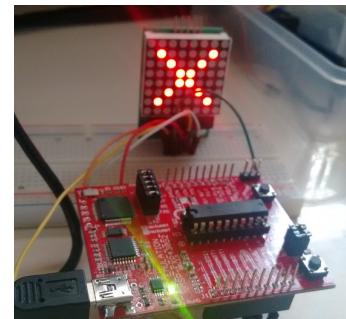


Fig. 13. Função parada() em funcionamento. Fonte: Autores

## REFERENCES

- [1] G1, “Ciclista morre após ser atropelado e arrastado em SP”. Disponível em: <http://g1.globo.com/sao-paulo/noticia/ciclista-morre-apos-ser-atropelado-e-arrastado-em-sp.ghtml>.
- [2] G1, “Brasil tem, em média, 32 ciclistas internados por dia devido a acidentes”. Disponível em: <http://g1.globo.com/bom-dia-brasil/noticia/2017/03/brasil-tem-em-media-32-ciclistas-internados-por-dia-devido-acidentes.html>.
- [3] DNIT, “NÚMERO DE VITIMADOS ENVOLVIDOS POR TIPO DE USUÁRIO”, 2011.
- [4] Mesriani Law Group, “Safety Tips to Avoid Bicycle Accidents”. Disponível em: <https://www.hg.org/article.asp?id=7752>.
- [5] Marc Lindsay, “10 Cycling Hand Signals You Need to Know”. Disponível em: <http://blog.mapmyrun.com/10-cycling-hand-signals-need-know/>.



Fig. 11. Função setaEsquerda() em funcionamento. Fonte: Autores