



UNIVERSIDADE FEDERAL DO PARÁ

Projeto de Semáforo Automatizado

Arthur Neves Moreira -201706840014

Elizabeth Ferreira Oliveira -201606840010

Marcos Paulo Braga Moreira -201406840071

Iglan Monteiro Miranda Cardeal - 201306840020

Belém -2019

Sumário

1.Introdução

2.Problemática

3. Cenário.

4.Solução

5.Orçamento

6.Conclusão

7.Referências

Anexo1:Código fonte

Anexo 2: Site

1.Introdução

Este projeto tem como finalidade aplicar o conhecimento adquirido na disciplina Circuitos elétricos , criando um produto que possa ser utilizado e aceito no mercado.

2.Problemática

Realizar implementação e teste de um protótipo de um sistema semafórico (em escala reduzida) para pedestres/veículos e adequado para Pessoas com Deficiências (PcDs).

3. Cenário.

O palco utilizado para o desenvolvimento do projeto foi uma via que possui dois sentidos(mão dupla) .Na qual devido ao fluxo, há necessidade de uma estrutura para que as pessoas possam atravessar com segurança.



Imagem 1: Via com dois sentidos.

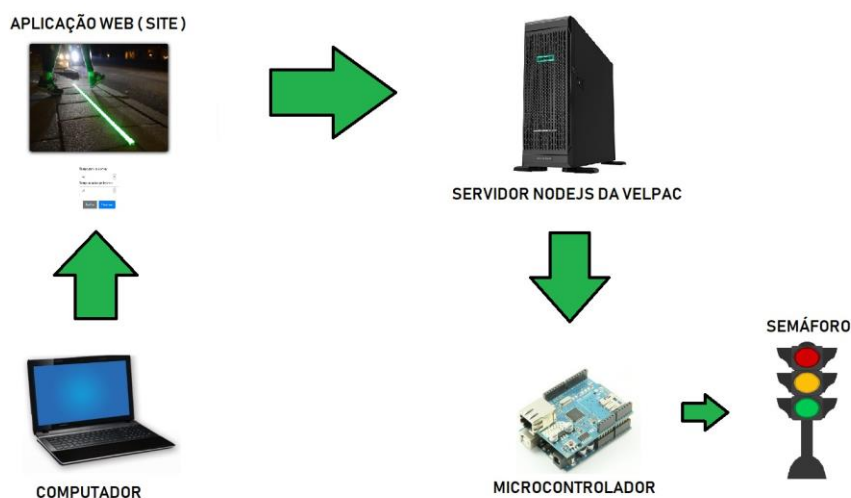
4.Solução

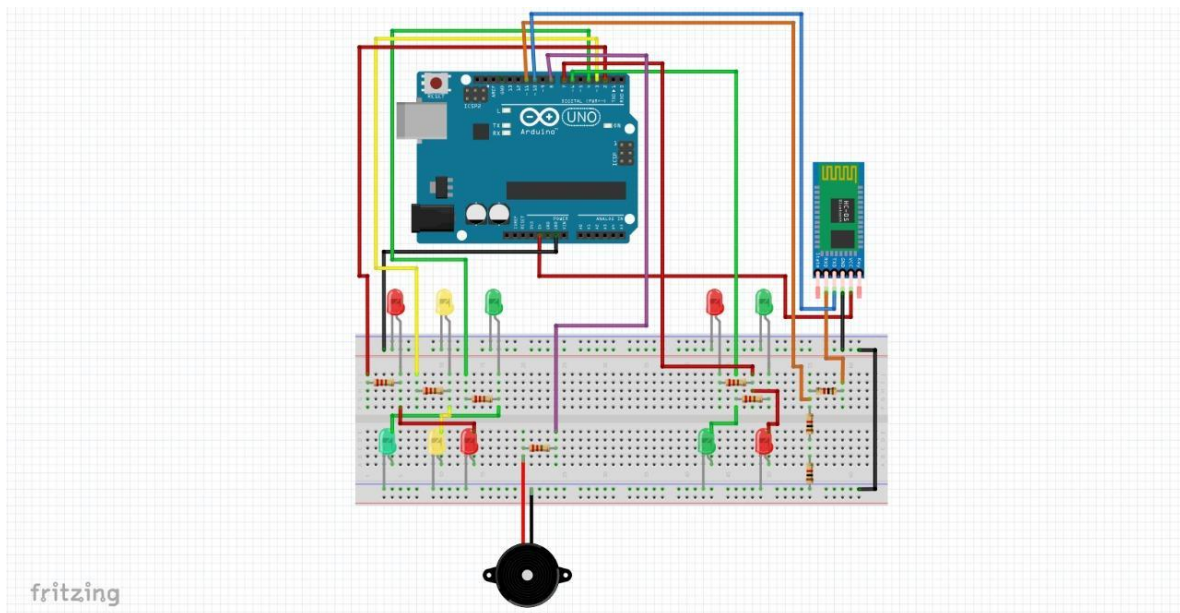
O gerenciamento do tempo de semáforo se dar através de um interface de usuário em uma aplicação web (site de gerenciamento de semáforos da velpac - SGS) onde, após o administrador se autenticar ele terá acesso ao sistema onde será possível determinar o

tempo das rotinas, cadastrar um semáforo, editar tempo limites de rotinas e adicionar novos administradores ao sistema.

Para demonstrar melhor, foi criado um protótipo em escala reduzida, onde temos uma maquete que representa um cruzamento de uma via com uma faixa de pedestre. Foi utilizado a plataforma de prototipagem eletrônica de hardware livre Arduino, com auxílio de componentes eletrônicos como jumpers, resistores, buzz sonoro, módulo de bluetooth para acionamento de botão acessível de pedestres e leds para montagem do protótipo, além da IDE do Arduino para exibição de informações relevantes e carregamento do código da aplicação, sendo que este código estará disponível para download. Logo a seguir, tem disponibilizado imagens e especificações do componentes usados no projeto.

Através do site gerenciador, ao submeter uma nova rotina, os dados serão enviados para o servidor da *VELPAC* que fica responsável por validar os dados e estabelecer a comunicação com o Arduino. Se tudo ocorrer como esperado, os dados serão enviados em formato de string com a seguinte estrutura: "**[tempo para os carros]|[tempo para os pedestres]**". No nosso exemplo, quando enviamos o tempo de 40 segundos para carros e 20 segundos para pedestres, o Arduino receberá a string "40|20" e o código do mesmo está configurado para detectar o recebimento destes dados do servidor e realizar as tratativas para aplicar os tempos no semáforo.





A exemplo de como imaginamos temos abaixo a imagem dos semáforos na via dupla , que tem como objetivo facilitar a travessia em avenidas muito movimentadas.



Imagem 3:Os semáforos na via de dois sentidos.

5.Orçamento

Materiais	Quantidade	Preço unitário	Preço Total
Arduíno Uno	1	R\$54,90	R\$54,90
Led vermelho	4	R\$0,30	R\$1,20
Led verde	4	R\$0,30	R\$1,20
Led Amarelo	2	R\$0,30	R\$0,60
Buzzer	1	R\$0,90	R\$0,90
Resistores	7	R\$0,30	R\$2,10
Caixa	1	R\$5,00	R\$5,00
Total			R\$65,90

6.Conclusão

A implantação de semáforos inteligentes não apenas facilita a travessia , proporciona segurança e inclusão social .

7.Referências

Imagem 1: <https://rnews.com.br/rua-sao-luis-pode-passar-a-ter-mao-unica-de-trafego.html>

Imagem 3: site: <https://dnsul.com/2018/geral/toda-extensao-da-rua-jose-cechinel-passa-ser-mao-dupla/>

Anexo1:Código fonte

```

#include <SoftwareSerial.h>
SoftwareSerial bluetooth(10, 11); // RX, TX

// Define os pinos que serao utilizados
int pedVerde = 6;
int pedVermelho = 7;
int carroVerde = 4;
int carroAmarelo = 3;
int carroVermelho = 2;

int buzzer = 8;

// delay padrao para carros e pedestres
double delayVerdeCarros = 10000;
double delayVerdePedestres = 5000;
int delayAmareloCarros = 3000;
int delayAmareloPedestres = 5000;

int delayAntesBotao;

// strings para tratar os dados vindo do servidor NodeJS
String dadosServidorNode, part1, part2;

void setup() // Define os pinos como saidas
{
    Serial.begin(9600);
    bluetooth.begin(9600);
    pinMode(pedVerde, OUTPUT);
    pinMode(pedVermelho, OUTPUT);
    pinMode(carroVerde, OUTPUT);
    pinMode(carroAmarelo, OUTPUT);
    pinMode(carroVermelho, OUTPUT);

    pinMode(buzzer, OUTPUT); // aviso sonoro

    digitalWrite(carroVerde, HIGH); // Coloca na posição inicial. Somente o verde dos carros
    digitalWrite(carroVerde, LOW);
    digitalWrite(carroVerde, LOW);
    digitalWrite(pedVerde, LOW);
    digitalWrite(pedVermelho, HIGH);
}

void loop()
{
    delayAntesBotao = delayVerdeCarros / 2;
    digitalWrite(carroVerde, HIGH); // Acende o verde dos carros e o vermelho dos pedestres
    void loop()
    {
        delayAntesBotao = delayVerdeCarros / 2;
        digitalWrite(carroVerde, HIGH); // Acende o verde dos carros e o vermelho dos pedestres
        digitalWrite(pedVermelho, HIGH);

        delay(delayAntesBotao);

        if (bluetooth.available() > 0) {
            char bluebyte = bluetooth.read();
            if (bluebyte == 'B') { // se o botao for apertado antes de ter passado metade do tempo
                Serial.println("Bluetooth ativado!");
                Serial.println("Espera somente metade do tempo dos carros.");
                digitalWrite(carroVerde, LOW);
                digitalWrite(carroAmarelo, HIGH); // apaga o verde dos carros e acende o amarelo
                delay(delayAmareloCarros); // aguarda mais 3 segundos

                digitalWrite(carroAmarelo, LOW); // apaga o amarelo dos carros e acende o vermelho
                digitalWrite(carroVermelho, HIGH);
                delay(1000);
                digitalWrite(pedVermelho, LOW); // apaga o vermelho dos pedestres e acende o verde
                digitalWrite(pedVerde, HIGH);

                int tempoPedestreLoop = delayVerdePedestres / 1000;
                for (int y = 0; y < tempoPedestreLoop; y++) {
                    digitalWrite(buzzer, HIGH);
                    delay(500);
                    digitalWrite(buzzer, LOW);
                    delay(500);
                }

                // delay(delayVerdePedestres); // aguarda mais 5 segundos

                digitalWrite(pedVerde, LOW);

                int delayDentroLoop = delayAmareloPedestres / 1000;
                digitalWrite(buzzer, HIGH); // buzz fica ligado para indicar que o amarelo de pedestre
                for (int x = 0; x < 5; x++) { // Pisca o vermelho dos pedestres
                    digitalWrite(pedVermelho, HIGH);
                    delay(500);
                    digitalWrite(pedVermelho, LOW);
                    delay(500);
                }
                digitalWrite(buzzer, LOW); // apos fechar o sinal pedestre, desliga o buzz
                digitalWrite(pedVermelho, HIGH);
            }
        }
    }
}

```

```

void loop()
{
    delayAntesBotao = delayVerdeCarros / 2;
    digitalWrite(carroVerde, HIGH); // Acende o verde dos carros e o vermelho dos pedestres
    digitalWrite(pedVermelho, HIGH);

    delay(delayAntesBotao);

    if (bluetooth.available() > 0) {
        char bluebyte = bluetooth.read();
        if (bluebyte == 'B') { // se o botao for apertado antes de ter passado metade do tempo
            Serial.println("Bluetooth ativado!");
            Serial.println("Espera somente metade do tempo dos carros.");
            digitalWrite(carroVerde, LOW);
            digitalWrite(carroAmarelo, HIGH); // apaga o verde dos carros e acende o amarelo
            delay(delayAmareloCarros); // aguarda mais 3 segundos

            digitalWrite(carroAmarelo, LOW); // apaga o amarelo dos carros e acende o vermelho
            digitalWrite(carroVermelho, HIGH);
            delay(1000);
            digitalWrite(pedVermelho, LOW); // apaga o vermelho dos pedestres e acende o verde
            digitalWrite(pedVerde, HIGH);

            int tempoPedestreLoop = delayVerdePedestres / 1000;
            for (int y = 0; y < tempoPedestreLoop; y++) {
                digitalWrite(buzzer, HIGH);
                delay(500);
                digitalWrite(buzzer, LOW);
                delay(500);
            }

            // delay(delayVerdePedestres); // aguarda mais 5 segundos

            digitalWrite(pedVerde, LOW);

            int delayDentroLoop = delayAmareloPedestres / 1000;
            digitalWrite(buzzer, HIGH); // buzz fica ligado para indicar que o amarelo de pedestre
            for(int x = 0; x<5; x++) { // Pisca o vermelho dos pedestres
                digitalWrite(pedVermelho, HIGH);
                delay(500);
                digitalWrite(pedVermelho, LOW);
                delay(500);
            }
            digitalWrite(buzzer, LOW); // apos fechar o sinal pedestre, desliga o buzz
            digitalWrite(pedVermelho, HIGH);

            digitalWrite(carroVermelho, HIGH);
            delay(1000);
            digitalWrite(pedVermelho, LOW); // apaga o vermelho dos pedestres e acende o verde
            digitalWrite(pedVerde, HIGH);

            int tempoPedestreLoop = delayVerdePedestres / 1000;
            for (int y = 0; y < tempoPedestreLoop; y++) {
                digitalWrite(buzzer, HIGH);
                delay(500);
                digitalWrite(buzzer, LOW);
                delay(500);
            }

            // delay(delayVerdePedestres); // aguarda mais 5 segundos

            digitalWrite(pedVerde, LOW);

            int delayDentroLoop = delayAmareloPedestres / 1000;
            digitalWrite(buzzer, HIGH); // buzz fica ligado para indicar que o amarelo de pedestre es
            for(int x = 0; x<5; x++) { // Pisca o vermelho dos pedestres
                digitalWrite(pedVermelho, HIGH);
                delay(500);
                digitalWrite(pedVermelho, LOW);
                delay(500);
            }
            digitalWrite(buzzer, LOW); // apos fechar o sinal pedestre, desliga o buzz
            digitalWrite(pedVermelho, HIGH);

            delay(1000);
            digitalWrite(carroVermelho, LOW);
            digitalWrite(carroVerde, HIGH);
            // quando o arduino receber uma nova rotina vinda do servidor NodeJS
            if (Serial.available() > 0) {
                while (Serial.available() > 0) {
                    dadosServidorNode = Serial.readString();
                }
                part1 = dadosServidorNode.substring(0, dadosServidorNode.indexOf("|"));
                part2 = dadosServidorNode.substring(dadosServidorNode.indexOf("|") + 1);

                delayVerdeCarros = part1.toDouble() * 1000;
                delayVerdePedestres = part2.toDouble() * 1000;
                Serial.println("Novos dados do servidor: " + String(dadosServidorNode) + "\nAplicando m
                Serial.println("Novo tempo para carros: " + String(delayVerdeCarros));
                Serial.println("Novo tempo para pedestres: " + String(delayVerdePedestres));
            }
        }
    }
}

```


Anexo 2: Site

<https://velpac.herokuapp.com/>