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.Simulação

6.Orçamento

7.Conclusão

8.Referências

Anexo 1:Código fonte

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

Nossa proposta é a implantação de dois semáforos para carros e dois semáforos para pedestres , utilizando recursos auditivos para informa a abertura e fechamento do sinal, viabilizando as pessoas com deficiências visuais sempre compreenderem se o sinal está prestes a fechar.

Será empregado a placa microcontroladora Arduino Uno, na qual fará o controle dos semáforos, o código utilizado encontra-se no anexo 1.

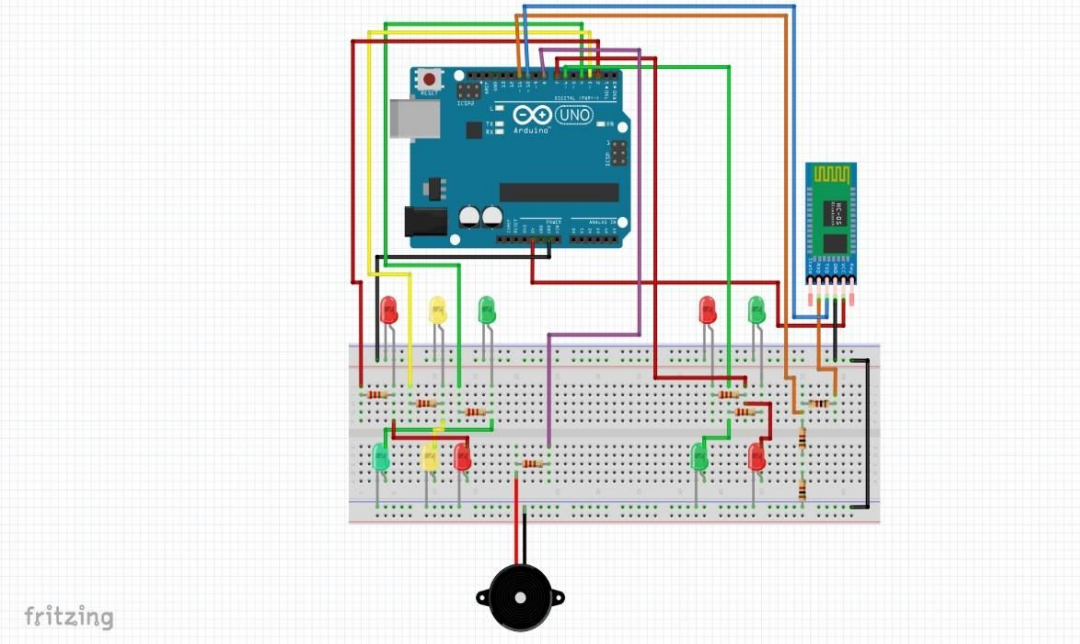

Imagem 2:Esquemático

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 melhor demonstrar , 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 está disponível para download em *https://velpac.herokuapp.com/*. 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.

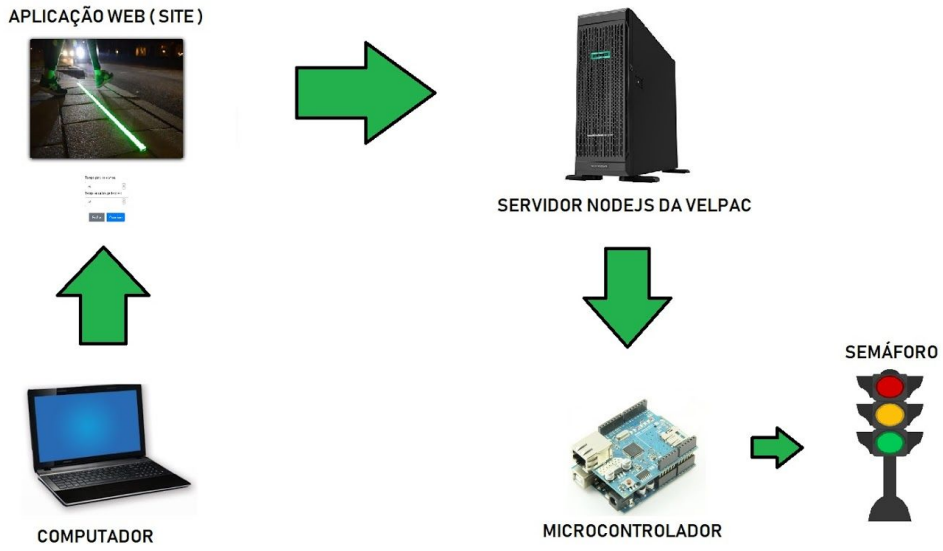

Imagem 3:Fluxo de acesso.

5.Simulação

Foi realizada uma simulação para demonstrar na prática a aplicação tendo efeitos no semáforo. Com o Arduino montado e os componentes do circuito inseridos na protoboard, conectado ao computador via porta serial, foi exibida a interface wew, na página de administração, foi definidos os tempos de teste para observarmos os efeitos no protótipo. Intervalos curtos de 4 segundos e 2 segundos e 40 segundos e 20 segundos para dinamizar os efeitos da gerência do semáforo.

Na simulação, temos um buzzer para favorecer a acessibilidade para deficientes, onde quando o semáforo para pedestres estiver fechado (vermelho pedestre aceso), o buzzer não emite som algum. Quando o semáforo estiver aberto para os pedestres (verde pedestre aceso), o buzzer irá emitir um bip intervalado, indicando que o sinal está aberto. Quando o semáforo estiver em estado de atenção (vermelho pedestre piscando), o buzzer irá emitir um bip continuo, indicando uma eminência de que o semáforo de pedestre irá fechar.

Nesta simulação, a via teórica onde o semáforo se encontra, possui um limite máximo de 40km/h, logo foi definido um tempo de alerta para os carros (semáforo amarelo) padrão de 3 segundos, e este valor não muda com alteração de rotina dos semáforos. Para os pedestres, o tempo de semáforo em alerta (vermelho pedestre piscando), foi definido o tempo padrão de 5 segundos e este valor também não muda com possível alteração de rotina dos semáforos.

A alteração de estado de vermelho para carros e verde para pedestres, não ocorre instantaneamente. Quando o sinal de carros fecha, o sinal verde de pedestre demora 1 segundo para abrir. A alteração de estado de vermelho para pedestre e verde para carros também não ocorre de imediato, pois o sinal de carros irá demorar 1,5 segundos para abrir depois do de pedestres fechar. Estes atrasos de mudança de estado são intencionais nesta simulação para efeitos de segurança pública.

6.Orçamento

Materiais	Quantidade	Preço unitário	Preço Total
Arduíno Uno	**1**	**R$54,90**	**R$54,90**
Led vermelho	**4**	**R$0,50**	**R$2,00**
Led verde	**4**	**R$0,50**	**R$2,00**
Led Amarelo	**2**	**R$0,50**	**R$1,00**
Buzzer	**1**	**R$0,90**	**R$0,90**
Resistores 1k ohm	**3**	**R$0,30**	**R$0,90**
Resistores 220 ohm	**4**	**R$0,30**	**R$1,20**
Madeira	**1 folha**	**R$10,00**	**R$10,00**
Jumpers	**1 pacote**	**R$6,00**	**R$6,00**
Módulo Bluetooth	**1**	**R$28,00**	**R$28,00**
Pregos	**1 pacote**	**R$5,00**	**R$5,00**
Total			**R$111,90**

7.Conclusão

O avanço na informatização e a automação de controle, trazem evolução nos métodos tradicionais utilizados . A elaboração de sistema de controle computadorizado centralizado, no qual um programa comanda diretamente o planejamento dos semáforos, por meio de pré programação, indicando a cada instante qual a situação luminosa que deve ocorrer já é realidade em algumas cidades.A implantação desses sistemas inteligentes não apenas facilita a travessia , proporciona segurança e inclusão social para todos .

8.Referências

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

Anexo 1: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 e o vermelho dos pedestres acesos
  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
  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 verde para carros, altera o estado para priorizar o verde de pedestre.
      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);
      }
```

```
      digitalWrite(pedVerde, LOW);

      int delayDentroLoop = delayAmareloPedestres / 1000;
      digitalWrite(buzzer, HIGH); // buzz fica ligado para indicar que o amarelo de pedestre esta ativado
      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);
      return; // ao fim, volta ao loop inicial.
    }
  }

  delay(delayAntesBotao);

  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 esta ativado
  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 nova rotina!");
    Serial.println("Novo tempo para carros: " + String(delayVerdeCarros));
    Serial.println("Novo tempo para pedestres: " + String(delayVerdePedestres));
  }
```