ESTUDO E IMPLEMENTAÇÃO DE UM SISTEMA PARA GERENCIAMENTO DE SEMÁFOROS PARA AMBULÂNCIAS EM SITUAÇÕES EMERGENCIAIS

Fabricio Augusto Giese[1]; Frederico Eduardo dos Santos[2]; Henrique Ferreira Almeida[3]; Marcelo Pieroni Martins; [4] Leandro de Campos Cardoso[5]

Moisés Henrique Ramos Pereira[6] - Orientador

Centro Universitário de Belo Horizonte, Belo Horizonte, MG

[1]fabgiese@gmail.com; [2]fredlu01@hotmail.com; [3]henriqueferreira2009@gmail.com; [4]pieroni14@hotmail.com; [5]leandrocamposcardoso@gmail.com; [6]moises.ramos@unibh.com

RESUMO: ESTE TRABALHO CONSISTE NA CRIAÇÃO E CONSTRUÇÃO DE UM SISTEMA INOVADOR DE CONTROLE DE SEMÁFOROS COM O FIM DE DAR PREFERÊNCIA PARA VEICULOS EMERGÊNCIAIS ATRAVÉS DE SENSORES ASSIM REDUZINDO O NUMERO DE ACIDENTES E TAMBEM DIMINUINDO O TEMPO DE DESLOCAMENTO DESTES VEÍCULOS.

ABSTRACT: THIS PROJECT RELAYS ON THE CREATION AND CONSTRUCTION OF AN INNOVATIVE TRAFFIC LIGHT CONTROL SYSTEM FOR THE PURPOSE OF GIVING PREFERENCE TO EMERGENCY VEHICLES THROUGH SENSORS THEREFORE REDUCING THE NUMBER OF ACCIDENTS AND ALSO DECREASING THE TIME SHIFTING OF THESE VEHICLES.

PALAVRAS-CHAVE: SISTEMA DE CONTROLE, SEMÁFOROS, TRÁFEGO, AMBULÂNCIAS, EMERGÊNCIA, MOBILIDADE URBANA

1 INTRODUÇÃO

O crescimento populacional projetado pelo IBGE trouxe e trará impactos significativos no tráfego nos grandes centros urbanos de todo o mundo e, com isso, as cidades vêm enfrentando problemas de mobilização urbana. Os semáforos são equipamentos de sinalização para ajudar na organização e comunicação de veículos em cruzamento de vias arteriais e na segurança da passagem dos pedestres. Eles trabalham em um sistema cronometrado com intervalo de tempo entre eles.

Nestes intervalos podem acontecer problemas com a passagem dos veículos preferenciais, que são os destinados a socorro e salvamento, os policiais, os de fiscalização e operações de trânsito e as ambulâncias, que tem preferência de passagem e podem ultrapassar o sinal vermelho quando estiverem com o alarme sonoro e as luzes vermelhas intermitentes ligadas, conforme o Art. 29 da lei de transito Brasileiro (CTBDigital,2014). Muitas vezes, o transito está em um nível de congestionamento crítico, impossibilitando dos carros poderem liberar a passagem destes veículos ou muitas vezes ele ultrapassa o sinal vermelho e se conflita com outro carro como ocorre diariamente em muitas cidades no mundo (G1, 2014).

1.1 DEFINIÇÃO DO PROBLEMA

No contexto citado acima, este artigo visa responder o seguinte questionamento: “É possível criar e construir um dispositivo que ao ser instalado tem como objetivo principal fazer com que veículos preferenciais cheguem mais rápido e com mais segurança aos seus destinos, em horários de grande tráfego sem causar acidentes de transito?”.

1.2 OBJETIVOS

Temos como objetivo principal propor um sistema que será demonstrado em um protótipo onde será instalado sensores emissores que serão colocados em cada ambulância e sensores receptores em cada proximidade do semáforo para saber quando um

veículo preferencial se aproxima. Quando ocorre a aproximação, o sensor ativa o sistema que por sua vez gerencia os semáforos e libera o fluxo da via onde este veículo se encontra, ao mesmo tempo ativa a proibição de passagem nos outros semáforos das vias laterais. Assim pretendemos diminuir o risco de outro veículo se conflitar com o veículo preferencial. Após a passagem do veículo preferencial, quando o mesmo sair da área de alcance do sensor, os semáforos voltam para os intervalos normais temporizados.

O presente trabalho tem como objetivo analisar como o transito e como os motoristas estão atrapalhando a passagem de veículos especiais e será desenvolvido a partir de várias pesquisas e observações. Assim, após a conclusão de todas as pesquisas e experimentos temos como objetivo principal criar um protótipo de mobilidade urbana para o controle de semáforos, e também uma maquete em escala reduzida, assim facilitando o entendimento do projeto e fazendo a demonstração da passagem de veículos em situações de emergência. O projeto conta também com possibilidade de futuras melhorias.

1.3 MOTIVAÇÃO

A motivação que nos levou a desenvolver este projeto vem do grande aumento dos congestionamentos e a dificuldade de mobilidade urbana, tanto para carros convencionais quanto para os emergenciais, além do 'aumento cada vez maior da frota de veículos circulando nas ruas das principais capitais' (Resende 2009; Souza 2009). Assim, visando diminuir ao máximo o tempo da rota e também visando evitar acidentes que muitas vezes acontecem pelo veículo ultrapassar o sinal vermelho e se confrontar com outro veículo.

2 REVISÃO DE LITERATURA

2.1 LEGISLAÇÃO

Como prevê no artigo 29 da Lei Brasileira de Transito, os veículos destinados a socorro de incêndio e salvamento, os de polícia, os de fiscalização e operação de trânsito e as ambulâncias, além de prioridade de trânsito, gozam de livre circulação, estacionamento e parada, quando em serviço de urgência e devidamente identificados por dispositivos regulamentares de alarme sonoro e iluminação vermelha intermitente (CTBDigital,2014).

Tomando como base a legislação, estes veículos em situações de emergência (com a sirene ligada) poderão transcender qualquer infração de transito inclusive passar em sinal vermelho.

2.2 TRABALHOS RELACIONADOS

Com uma análise feita sobre congestionamento de transito, segundo Resende (2009) e Sousa (2009), o problema de mobilidade urbana reflete sobre o possível travamento total do transito, com o aumento próximo de 100 mil carros por ano nas principais capitais brasileiras. Isso é um problema que poderá impossibilitar a circulação de ambulâncias nas principais vias brasileiras. Conviver com níveis de congestionamento alto é uma realidade para a maioria das metrópoles mundiais, mas a paralisação dos fluxos de trânsito deve ser evitada a todo custo, e a tendência é que continue aumentando, pois a frota está em constante expansão. (Resende 2009; Souza 2009).

O presente trabalho tem uma proposta parecida com o projeto de Sharma (2013) que propõe o controle de semáforo para veículos de emergência usando tecnologia RFID. Por meio de sensores de rádio frequência, ele consegue identificar a aproximação

do veículo e considerar a densidade do tráfico para assim fazer o controle dos semáforos podendo facilitar a circulação deste veículo. Porém o projeto proposto usará sensor de presença no veículo e na via e, com a aproximação dos sensores, fará um controle nos semáforos para o fluxo da via agir a favor do veículo emergencial, em especial as ambulâncias.

Uma outra proposta que tentou também resolver a mesma problemática que a deste trabalho foi a tentativa do vereador de Recife, Dr. Rogério de Lucca, que propôs a criação de uma faixa exclusiva e preferencial para os veículos em situação de emergência. (Diário de Pernambuco 2011).

3 MATERIAIS E MÉTODOS

O desenvolvimento deste trabalho, além das pesquisas e observações, inclui também o desenvolvimento do software em conjunto com diferentes tipos de hardware que irão controlar os semáforos por meio de uma lógica orientada á objetos que proporciona uma maior reutilização do código em várias situações diferentes.
Os principais componentes de hardware a serem utilizados no nosso protótipo em pequena escala estão relacionados assim:

- Seis lâmpadas de LED multicoloridas de alto brilho.
- Seis resistores de 100k
- Placa Protoboard 640 furos
- Sensor Infravermelho Sharp GP2Y0A41SK0F (4-30cm)
- Arduino UNO R3

O código de software será desenvolvido em linguagem C em conjunto com o programa Arduino IDE 1.0.1 e suas bibliotecas em que serão utilizados 6 portas lógicas digitais conectadas entre si que farão o controle dos semáforos.

Após a análise completa do projeto e das fontes escolhidas e pesquisadas, percebe-se uma grande importância no relacionamento deste com outras áreas de conhecimento, que são de suma importância para o entendimento e construção de um projeto completo que abrange como tudo realmente acontece na vida real.

Primeiramente foi feito um estudo completo sobre a mobilidade urbana onde concluímos que realmente é viável e necessária a criação de novos conceitos para melhorar a mobilidade nos grandes centros, pois de acordo com o estudo feito por Paiva, C.(2007) na região metropolitana de São Paulo, o crescimento de veículos circulando por dia aumentou drasticamente nos últimos 10 anos.
Como pode-se ver na Tabela 1, o crescimento médio de carros á cada 10 anos tem uma média de 550 mil veículos.

Observando esse grande aumento de veículos nota-se a grande dificuldade de locomoção em horários críticos do dia, onde uma parcela alta de carros circula ao mesmo tempo assim causando congestionamentos massivos nos grandes centros. Segundo dados apresentados por Paiva, C. (2007) na Gráfico 1, a quantidade de veículos circulando dobra em horários de pico e fica em taxas bem baixas em horários na madrugada.

Relacionando as informações já citadas, a Tabela 2 apresenta as porcentagens do aumento de veículos nas décadas de 1987-1997 e 1997-2007 nos horários de maior fluxo.

A maior variação nos picos podem explicar os aumentos significativos dos congestionamentos nos períodos do dia apresentados. E pode-se notar que houve um maior aumento em todas as áreas analisadas na década de 1987-1997 foi quando ocorreu o ‘boom’ automobilístico. Como está

destacado na Tabela 2, somente no pico das 19hrs no período de 1997-2007 que o aumento foi maior que nos anos anteriores.

Usando conhecimentos de estatística foi feita uma análise de hipótese entre as mesmas regiões como consta no boxplot (gráfico de caixa) do Gráfico 2, apresentando os resultados que o centro expansivo não tem o mesmo potencial de trânsito que a Cidade de São Paulo e a Região Metropolitana de São Paulo, porém não se pode tirar conclusões estatísticas entre a cidade de São Paulo e a Região Metropolitana de São Paulo pelo fato de não conter valores aberrantes ao comparar os potenciais de cada região.

Número de veículos circulando por dia (incluem táxis, motos e caminhões)*. Origem e destino na região Metropolitana de São Paulo			
	OD 1987 veículos	**OD 1997 veículo**	**OD 2007 veículos**
Região Metropolitana	1272364	1853357	2370515
Cidade de São Paulo	1022230	1347731	1697755
Centro Expandido	667716	781622	946833
*Não incluem frota de empresas de serviço, distribuição de mercadorias, etc .			

Tabela 1 – Número de Veículos Circulando por dia. Paiva, C. (2007).

	Variação no número de autos circulando (não inclui táxi e outros) Origem e destino na região metropolitana de São Paulo			
	1987-1997		**1997-2007**	
	HPM (8Hrs) Var. % Autos	**HPT (19Hrs) Var. % Autos**	**HPM (8Hrs) Var. % Autos**	**HPT (19Hrs) Var. % Autos**
Região Metropolitana	52%	41%	21%	29%
Cidade de São Paulo	36%	27%	16%	23%
Centro Expandido	27%	14%	6%	15%

Tabela 2 – Variação do numero de autos circulando. Paiva, C. (2007).

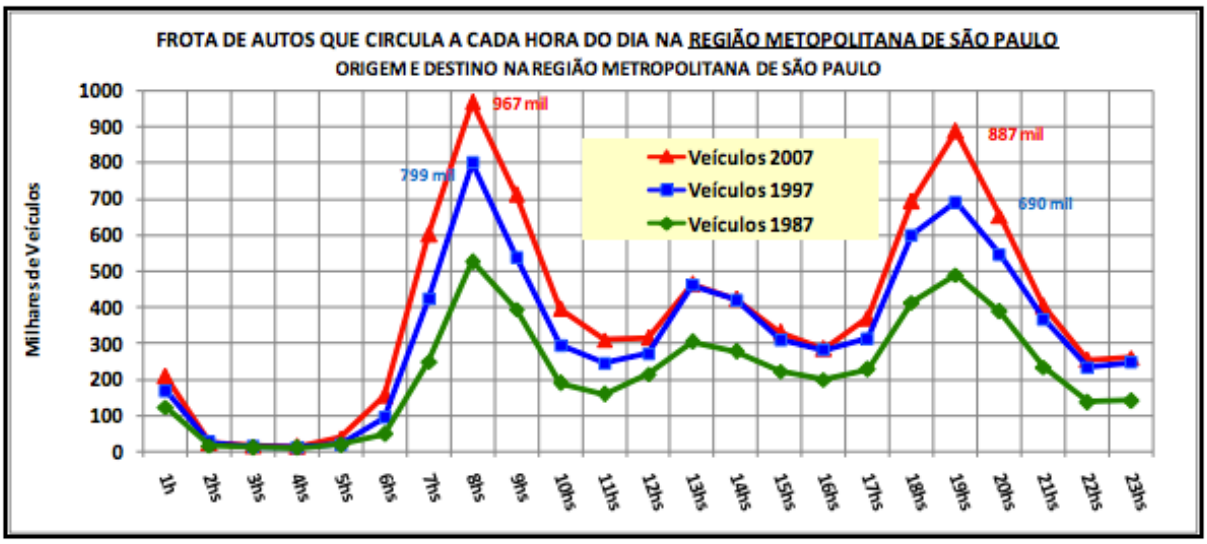


Gráfico 1 – Numero de Veículos Circulando por dia. Paiva, C. (2007).

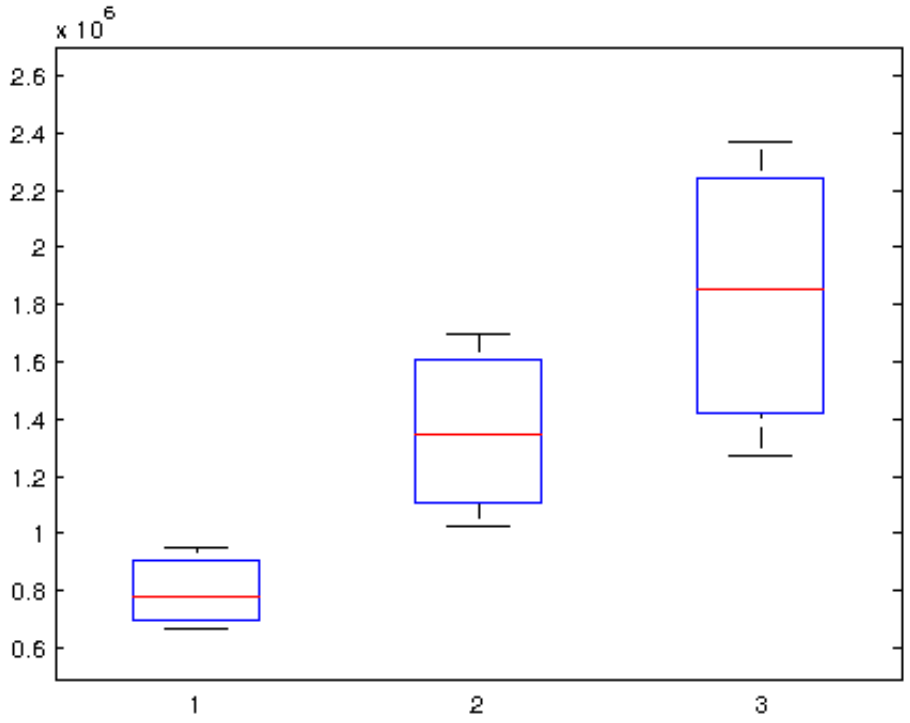


Gráfico 2: Análise de hipótese da quantidade de carros entre 1, 2 e 3.

Legenda: 1 – Centro Expandido de São Paulo, 2 – Cidade de São Paulo e 3 – Região Metropolitana de São Paulo.

Estudos feitos pela FGV(2012) demonstraram grandes custos que um congestionamento traz, e a diferença monetária entre ter e não ter congestionamento como podemos ver a tabela seguinte:

Itens	Sem congestionamento R$	Com Congestinamento R$	Variação R$	Variação x froto R$
Gasolina por carro/ano	1.608,42	2.483,52	875,1	3.948.575.185
Diesel por onibus/ano	43.350,02	57.221,55	13.871,54	206.880.087
Poluição por carro/ano	59,59	145,69	86,1	388.471.600
Poluição por onibus/ano	2.219,64	3.461,02	1.241,38	18.513.938
Transporte de carga (R$/Ton)	61,62	75,27	13,65	1.955.718.654
Total (R$)				6.518.159.464

Tabela 3: Custo de congestionamento na cidade de São Paulo

E as funções usadas para obter esses números foram:

Função de consumo de combustíveis pelos carros:

C = 0,09543 + 1,26643 / V – 0,00029 V

C = Consumo (l/km)

V = Velocidade (km/h)

Função de consumo de diesel pelos ônibus:

C = 0,44428 + 0,00008 V2 - 0,00708 V + 1,37911 / V + 0,00107 CR

V = velocidade (km/h)

C = Consumo (l / km)

CR = carregamento (sentado + em pé)

Custo do transporte de carga:

Custo-peso = [A + (B * X)],

A = custo de carregar e descarregar

B = custo de transferência

X = distância

A e B derivam das funções:

A= [(CF / H) * Tcd / CAP

B={[CF / (H * V)] + CV}/ CAP

Custo fixo (CF) = R$ 7.694,30

Horas trabalhadas (H) = 248,60 horas

Tempo de carga/descarga (Tcd) = 2,5 horas

Capacidade de carga do veículo (CAP) = 2,25

Custo variável por km (CV) = R$ 0,5001

4 ANÁLISE DE PROJETO

No desenvolvimento deste projeto foi criado um algoritmo na linguagem de programação C que funciona em conjunto com o Arduino UNO e os sensores descritos na sessão 3 (Materiais e Métodos) deste projeto.

A lógica de programação se descreve com primeiramente foram utilizadas seis portas lógicas

digitais para a ligação dos *LED's* e uma porta analógica para a ligação do sensor *SHARP*.

Para a definição do tempo de sequência entre o semáforo e o sensor foi utilizado a biblioteca *Thread,* onde foi determinado um *loop* que executa o controle da sequência dos dois 'semáforos' em que os LED's vermelho, amarelo e verde devem ser ligados ou desligados.

Assim foi obtida uma sequência de funções onde o controle de acender e apagar os LED's funcionam em sincronia e de acordo com as necessidades do projeto.

Foram criadas também algumas funções para que tudo funcione como esperado, em exemplo a isso estão as funções do sensor com uma sequência de lógica de trocas de semáforo e a função de mudança de sensor onde deixa o semáforo 1 verde e deixa o

vermelho o semáforo 2 e o semáforo 2 fica amarelo por 5ms.

E por fim algumas estruturas de condição (*if* e *else)* que por sua vez ativam o sensor para que ele capte o sinal do veículo, enviando assim o comando para que a troca dos semáforos seja feita de uma maneira que as condições pré-estabelecidas sejam respeitadas.

5 RESULTADOS EXPERIMENTAIS

Este protótipo tem os seguintes aspectos, os tempos dos semáforos são de 25s no sinal vermelho, 12,5s no sinal amarelo e 25s no sinal vermelho como mostra na tabela 3, e o sensor ele age da seguinte maneira, quando ele ativa demora 1ms para começar o processo e depois que o veículo passa ele demora 2000ms para voltar o ciclo dos sinais, como demonstra na tabela 4.

Depois foi feito um teste na via da ambulância com a via oposta a ela e a reação antes de depois do sensor captar como demonstra na tabela 5 e nela podemos ver que os ciclos funcionam perfeitamente.

	Tempo Sinal		
	Vermelho	**Amarelo**	**Verde**
Ms	2500	1250	2500
S	25	12,5	25

Tabela 3: Tempo de cada semáforo no ciclo.

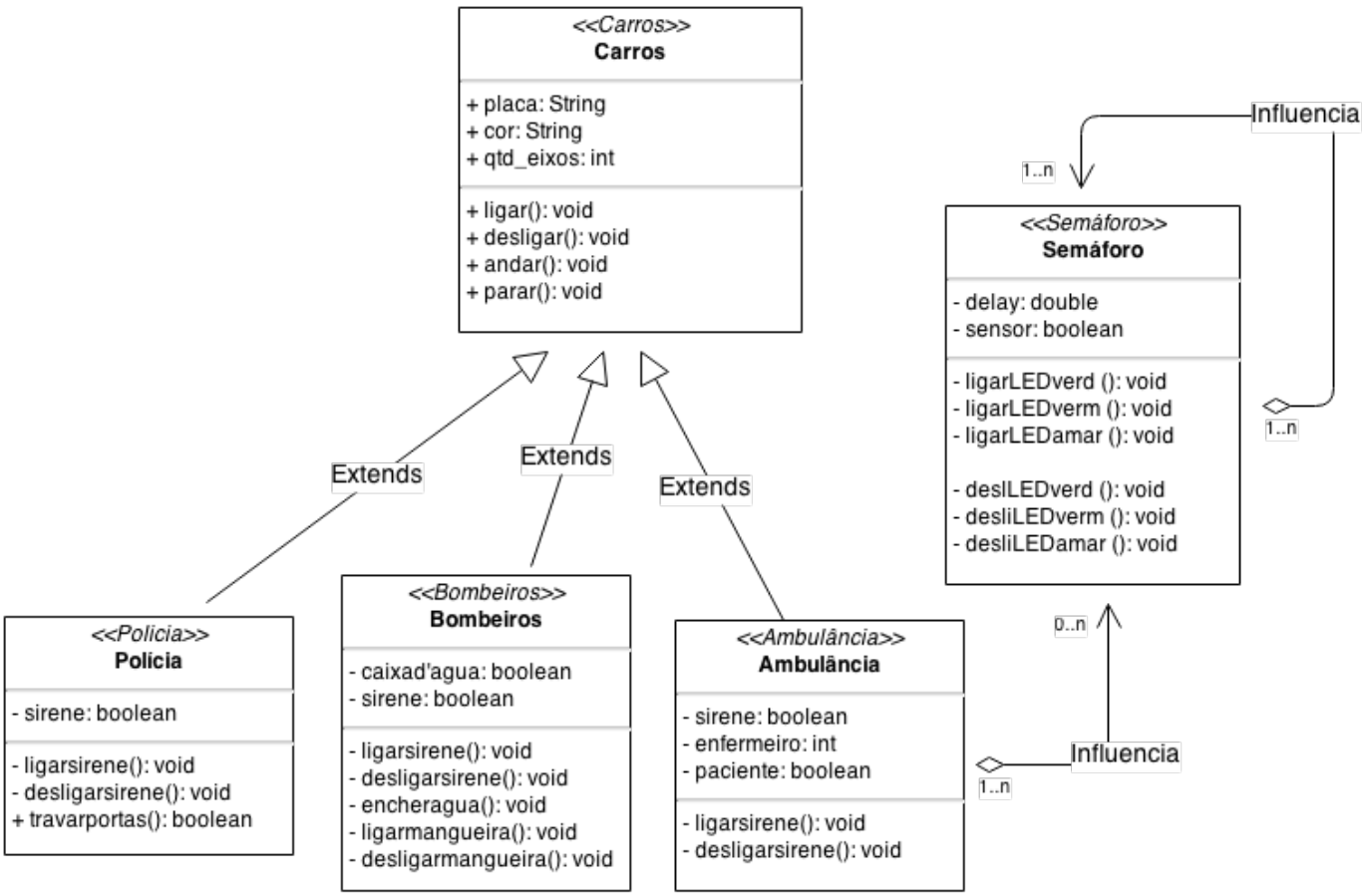
	Tempo Sensor	
	Ativação	**Desativação**
Ms	1	2000
S	0,01	2

Tabela 4: Tempo da ativação e desativação do sensor.

Ativação do sensor (Ciclo do Semáforo)			
Antes		**Depois**	
Ambulância	**Oposto**	**Ambulância**	**Oposto**
Verde	Vermelho	Vermelho	Verde
Vermelho	Verde	Vermelho	Verde
Vermelho	Amarelo	Verde	Vermelho
Amarelo	Vermelho	Vermelho	Amarelo

Tabela 5: Reações dos semáforos depois da ativação do sensor.

5.1 Modelagem – UML



6 CONCLUSÃO

Após o término deste projeto foi concluído que foi atingido com eficiência a sua finalidade, que é a de dar o suporte adequado na mobilidade urbana focando na passagem de veículos emergenciais.

Para isso foi criado um protótipo de semáforos e sensores de presença com o intuito de fazer o controle dele em casos de emergência, podendo assim dar um suporte e ao mesmo tempo ajudar na segurança dos condutores, pedestres e profissionais envolvidas nesse tipo de serviço.

Com isso concluímos este projeto com total satisfação por parte da equipe por ter atingido o seu objetivo principal, possibilitando assim planos futuros de melhoramentos e também para a implementação em escala real e com larga distribuição para todo o mundo.

7 APÊNDICE A

```
//Arduino UNO

//Código fonte

//Includes

#include <Thread.h>

#include <ThreadController.h>

#include <util/delay.h>


//Declaração de variaveis

//Thread

ThreadController controle;

Thread thread_sensor;

Thread semaforo;

//Constantes do semaforo 1

int s1_vermelho = 10; //Semaforo 1 - LED vermelho no pino 10

int s1_amarelo = 9; //Semaforo 1 - LED amarelo no pino 9

int s1_verde = 8; //Semaforo 1 - LED verde no pino 8

//Constantes do semaforo 2

int s2_vermelho = 7; //Semaforo 2- LED vermelho no pino 7

int s2_amarelo = 6; //Semaforo 2 - LED amarelo no pino 6

int s2_verde = 5; //Semaforo 2 - LED verde no pino 5

int pin_sensor = 4; //Led sensor

//Temporisador

unsigned long tempo = 2500; // Temporisador - Timer do LED Vermelho e Verde

unsigned long tempoAm = 1250; // Temporisador - Timer tempo do LED amarelo

int tempo_sensor = 1000; //Tempo de verificação do sensor

unsigned long tultimo; //Ultimo tempo verificado pelo millis

//Variaveis do sensor sharp

int sensorPin = 1; //Sensor Sharp - Entrada analogica 1

int lm_distancia = 80; //Distância de proximidade do sensor

int gdistancia = 100; //Variavel global de distância

int test = 0; //Variavel de controle


//Configurações

void setup() {

        thread_sensor.setInterval(tempo_sensor); //Define o tempo da thread

        thread_sensor.onRun(sensor); // Define a thread para executar a função sensor()

        semaforo.setInterval(1); //Define o tempo da thread

        semaforo.onRun(loop_led); // Define a thread para executar a função sensor()

        controle.add(&thread_sensor); //Adiciona a thread ao controlador

        controle.add(&semaforo); //Adiciona a thread ao controlador

        pinMode(s1_vermelho, OUTPUT); // Declarando que a variavel s1_vermelho é saída de dados
```

```
        pinMode(s1_amarelo, OUTPUT); // Declarando que a variavel s1_amarelo é saída de dados
        pinMode(s1_verde, OUTPUT); // Declarando que a variavel s1_verde é saída de dados
        pinMode(s2_vermelho, OUTPUT); // Declarando que a variavel s2_vermelho é saída de dados
        pinMode(s2_amarelo, OUTPUT); // Declarando que a variavel s2_amarelo é saída de dados
        pinMode(s2_verde, OUTPUT); // Declarando que a variavel s2_verde é saída de dados
        pinMode(pin_sensor, OUTPUT); // Declarando que a variavel pin_sensor  saida de dados
      Serial.begin(9600); //Abre a porta serial com a taxa de transferência de 9600bps

}

//Execução em loop(Main)
void loop() {
        controle.run(); //Executa a thread

}
//Funções
//Funções semaforo 1
void semaforo1_vermelho() {
  sensor();
        Serial.println("Semaforo 1 loop");
        Serial.println("Vm(1)A(0)Vd(0)");
        digitalWrite(s1_vermelho, 1);
        digitalWrite(s1_amarelo, 0);
        digitalWrite(s1_verde, 0);
}
void semaforo1_amarelo() {
  sensor();
        Serial.println("Semaforo 1 loop");
        Serial.println("Vm(0)A(1)Vd(0)");
        digitalWrite(s1_vermelho, 0);
        digitalWrite(s1_amarelo, 1);
        digitalWrite(s1_verde, 0);

}
void semaforo1_verde() {
  sensor();
        Serial.println("Semaforo 1 loop");
        Serial.println("Vm(0)A(0)Vd(1)");
        digitalWrite(s1_vermelho, 0);
        digitalWrite(s1_amarelo, 0);
        digitalWrite(s1_verde, 1);
}

//Funções semaforo 2
void semaforo2_vermelho() {
  sensor();
        Serial.println("Semaforo 2 loop");
        Serial.println("Vm(1)A(0)Vd(0)");
        digitalWrite(s2_vermelho, 1);
        digitalWrite(s2_amarelo, 0);
        digitalWrite(s2_verde, 0);

}
void semaforo2_amarelo() {
  sensor();
        Serial.println("Semaforo 2 loop");
        Serial.println("Vm(0)A(1)Vd(0)");
        digitalWrite(s2_vermelho, 0);
        digitalWrite(s2_amarelo, 1);
        digitalWrite(s2_verde, 0);
}
void semaforo2_verde() {
  sensor();
```

```
        Serial.println("Semaforo 1 loop");
        Serial.println("Vm(0)A(0)Vd(1)");
        digitalWrite(s2_vermelho, 0);
        digitalWrite(s2_amarelo, 0);
        digitalWrite(s2_verde, 1);
}

//Função mudança sensor
void muda_led() {
        //Função deixa o semaforo 1 verde e deixa vermelho o semaforo 2
        //Semaforo 2 - amarelo por 5 ms
        digitalWrite(pin_sensor, 1);
          digitalWrite(s2_vermelho, 0);
                digitalWrite(s2_amarelo, 1);
                digitalWrite(s2_verde, 0);
          digitalWrite(s1_vermelho, 1);
                digitalWrite(s1_amarelo, 0);
                digitalWrite(s1_verde, 0);
          delay(1500);
        Serial.println("STOP");
        while (gdistancia < lm_distancia) {
                float volts = analogRead(sensorPin) * 0.0048828125; // valor do sensor * (5/1024) - se estiver em 3.3.volts então muda de  5 para 3.3
                float distancia = 65 * pow(volts, -1.10);
                gdistancia = distancia;
                //Semaforo 2 - Vermelho
                digitalWrite(s2_vermelho, 1);
                digitalWrite(s2_amarelo, 0);
                digitalWrite(s2_verde, 0);
                //Semaforo 1 - verde
                digitalWrite(s1_vermelho, 0);
                digitalWrite(s1_amarelo, 0);
                digitalWrite(s1_verde, 1);
          Serial.println("Esperando ambulncia");
                delay(2000);
                test = 1;
          }
          Serial.println("GO");
       digitalWrite(pin_sensor, 0);
}
void sensor() {
        float volts = analogRead(sensorPin) * 0.0048828125; // valor do sensor * (5/1024) - se estiver em 3.3.volts então muda de  5 para 3.3
        float distancia = 65 * pow(volts, -1.10); //Formula para pegar a distancia
        gdistancia = distancia;
}

void loop_led(){
  if(gdistancia>lm_distancia){
  semaforo1_verde();
  semaforo2_vermelho();
  delay_ms(tempo);
}else{muda_led();}

if(gdistancia>lm_distancia){
  semaforo1_amarelo();
  semaforo2_vermelho();
    delay_ms(tempoAm);
}else{muda_led();}

if(gdistancia>lm_distancia){
      semaforo1_vermelho();
      semaforo2_verde();
      delay_ms(tempo);
}else{muda_led();}
//Loop S1 - vermelho / S2 - verde
if(gdistancia>lm_distancia){
  semaforo2_amarelo();
```

```
  semaforo1_vermelho();
  delay_ms(tempoAm);
}else{muda_led();}
if(gdistancia>lm_distancia){
    semaforo2_vermelho();
    semaforo1_verde();
 }else{muda_led();}
}
//Funcão para delay sem travar o
programa(delay/sensor)
void delay_ms(unsigned int time)
{
  while ((time--)&&(gdistancia>lm_distancia)){
    sensor();
    if(gdistancia>lm_distancia){
   _delay_ms(1);
   Serial.println(time/100);
    }
  }
```

8 REFERÊNCIAS

DINIZ, J. CTBDigital. Disponível em: <http://www.ctbdigital.com.br/?p=InfosArtigos&Registro=277&artigo=29>. Acessado em: 07 de setembro de 2014.

Portal G1. Disponível em: <g1.globo.com/são-paulo/são-jose-do-rio-preto-aracatuba/noticia/2014/04/acidente-entre-carro-e-ambulancia-deixa-tres-feridos-em-rio-preto.html>. Acessado em: 07 de setembro de 2014.

Diário de Pernambuco. Disponível em <http://blogs.diariodepernambuco.com.br/mobilidadeurbana/2011/10/abram-alas-para-as-ambulancias/> Acessado em: 15 de setembro de 2014.

SHARMA, Suresh; Traffic Light priority control for emergency vehicle using RFID. International Journal of Innovations in Engineering and Technology (IJIET) 2013.

RESENDE, Paulo; SOUZA, Paulo; Mobilidade Urbana nas grandes cidades brasileiras: um estudo sobre os impactos do congestionamento. SIMPOI ANAIS 2009.

PAIVA, C. Qual o número de veículos que circula em São Paulo? Disponível em PDF: <http://www.sinaldetransito.com.br/artigos/qual_o_numero_de_veiculos_em_sao_paulo.pdf> Acessado em: 01 de outubro de 2014.

EPUSP; Controle para semáforos de um cruzamento. PCS2011/2305/2355 - Laboratório Digital. Versão 2013.

JACEE 2012, Minicurso Arduino. Disponível em: < http://www.inf.ufes.br/~erus/arquivos/ERUS_minicurso%20arduino.pdf> Acessado em: 16 de outubro de 2014.

Programando com Threads em C. Disponível em: <http://www2.dcc.ufmg.br/disciplinas/aeds3_turmaN/pthreads.pdf> Acessado em: 01 de novembro de 2014.

FGV-EAESP, Os custos do congestionamento na cidade de São Paulo. Disponível em: < http://www.mobilize.org.br/midias/pesquisas/custos-do-congestionamento-em-sp.pdf> Acessado em: 27/11/2014.