

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Samuel Izaías Alves

**Semáforo inteligente para veículos de
emergência**

Monte Carmelo, MG

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Samuel Izaias Alves

Semáforo inteligente para veículos de emergência

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Monte Carmelo, MG

Samuel Izaías Alves

Semáforo inteligente para veículos de emergência

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Monte Carmelo, MG, 19 Dezembro de 2018:

Orientador
Prof. Dr. Jefferson Rodrigo de Souza

Prof. Dr. Leandro Nogueira Couto

Prof. Dr. Rodrigo Sanches Miani

Monte Carmelo, MG

Agradecimentos

Agradeço primeiramente a Deus pela oportunidade de poder crescer em sabedoria.

Agradeço muito ao meu Orientador Jefferson Rodrigo de Souza pelo imenso apoio.

Agradeço a toda minha família que estão sempre do meu lado.

E não poderia deixar de agradecer à minha futura esposa Sabrina Izabelle, que por todo tempo me incentivou a ir adiante em meus projetos.

Resumo

Diante da dificuldade de locomoção dos Veículos de Emergência (VE) no atual cenário caótico do trânsito brasileiro, foi desenvolvido o protótipo de um dispositivo, baseado na plataforma Arduino, o qual gerencia a passagem dos VE por cruzamentos onde há semáforos através de um sistema de comunicação com segurança. O VE quando aciona o dispositivo, esse sistema inteligente envia dados e o semáforo inteligente recebe essas informações, visando o tráfego do VE de forma segura e adequada. O resultado deste trabalho resultou em um protótipo que proporciona uma maior agilidade no atendimento das ocorrências solicitadas pelos VE's, assim produzindo um aumento significativo nas estatísticas das ocorrências bem atendidas ao longo dos anos.

Palavras-chave: Trânsito, Semáforo, Veículos de Emergência, Sensores.

Lista de ilustrações

Figura 1 – O transmissor MX-FS-03V opera na frequência de 433,92MHz e pode ser alimentado com tensão entre 3,5Vcc e 12Vcc. E a distância de transmissão é entre 20m à 200m (depende da tensão), tendo como velocidade de transmissão de dados: 4 Kb/s. (PAULA, 2015)	11
Figura 2 – Este receptor observa a presença de dois pinos dedicados a transmissão de dados. Além disso, ele opera na faixa de frequência de 433,92 MHz e deve ser alimentado com 5 Vcc (PAULA, 2015).	12
Figura 3 – A imagem do lado esquerdo representa o botão que pode ser acionado pelo pedestre atualmente. E a imagem do lado direito apresenta um infra-vermelho, onde após o pedestre cruzar o feixe, causa interrupção do receptor de luz infra-vermelho, simulando o acionamento do botão (CARDOSO, 2012).	13
Figura 4 – Visão ampla do sistema de presença (CARDOSO, 2012).	14
Figura 5 – Mapa gerado pelo sistema de aquisição (GALANTE; GARCIA, 2014).	15
Figura 6 – Maquete construída e desenvolvida para simular um cruzamento com dois semáforos inteligentes em funcionamento real com o Sistema Inteligente.	16
Figura 7 – Adicionado o sensor receptor na Maquete desenvolvida da segunda etapa.	17
Figura 8 – Semáforo Inteligente.	17
Figura 9 – ARDUINO UNO (ARDUINO, 2018).	19
Figura 10 – Módulo RF 433MHz (433MHZ, 2018).	19
Figura 11 – Módulo Relê (RELê, 2018).	21
Figura 12 – Esquema de ligação do módulo relê. (FLIPEFLOP, 2017)	22
Figura 13 – Carcaça de um semáforo real.	23
Figura 14 – Semáforo equipado com lâmpadas e relês.	23
Figura 15 – Teste de 5 metros, transmissor sem antena, de dentro do carro.	24
Figura 16 – Teste de 5 metros, receptor (Semáforo) sem antena, ambiente aberto.	24
Figura 17 – Teste de 10 metros, com antena, ambiente aberto.	25
Figura 18 – Semáforo com teste de todas a lâmpadas.	26
Figura 19 – Inicialização do semáforo inteligente.	27
Figura 20 – Transmissor emitindo sinal de trânsito livre.	27
Figura 21 – Sinal verde durante fluxo normal.	28
Figura 22 – Sinal Amarelo durante fluxo normal.	28
Figura 23 – Sinal vermelho durante fluxo normal.	29
Figura 24 – Transmissor enviando requisição de abertura do sinal através da emergência.	29

Figura 25 – Semáforo em estado de atenção.	30
Figura 26 – Semáforo em estado livre.	30
Figura 27 – Declaração de variáveis do TRANSMISSOR.	35
Figura 28 – Função de configuração.	35
Figura 29 – Função que executa o fluxo que ficará em Loop.	36
Figura 30 – Continuação da função Loop.	36
Figura 31 – Declaração de variáveis do RECEPTOR.	37
Figura 32 – Função de configuração.	37
Figura 33 – Função Loop.	38
Figura 34 – Função de acionamento dos relês.	38
Figura 35 – Continuação da função de acionamento dos relês.	39
Figura 36 – Fluxo do normal do semáforo.	39
Figura 37 – Função que faz a transição do sinais para emergência.	40
Figura 38 – Função emergência.	40

Lista de abreviaturas e siglas

VE	Veículos de Emergência
CTB	Código de Trânsito Brasileiro
RF	Rádio Frequência
TCC	Trabalho de Conclusão de Curso

Sumário

1	INTRODUÇÃO	9
2	TRABALHOS RELACIONADOS	11
2.1	Comunicação de rádio frequência usando transmissor/receptor	11
2.2	O sensor de presença para os pedestres em semáforos	13
2.3	Aquisição de dados usando os sensores de baixo custo	14
3	METODOLOGIA	16
3.1	Construção da Maquete	16
3.2	Implantação de Sensores	16
3.3	Construção do Semáforo Real	17
3.4	Hardware	18
3.5	Plataforma ARDUINO	18
3.6	Sensor de Rádio Frequência RF433	19
3.6.1	Pinagem dos Módulos TX/RX de 433 MHz	20
3.6.2	Especificações dos Módulos TX/RX de 433 MHz	20
3.6.3	Antena	20
3.6.4	Alcance	21
3.7	Modulo relê	21
3.7.1	Esquema de Ligação do Módulo Relê	22
4	RESULTADOS	23
4.1	Construção do semáforo	23
4.2	Teste de 5 metros	24
4.3	Teste de 10 Metros	25
4.4	Teste Final	26
4.4.1	Início do sistema	26
4.4.2	Trânsito Livre	26
4.4.3	Trânsito em Emergência	26
	Conclusão	31
	REFERÊNCIAS	32
	APÊNDICES	34

1 Introdução

Historicamente, compreendemos que a revolução industrial desencadeou um crescimento populacional nas metrópoles brasileiras, dificultando um bom planejamento de urbanização. E ainda, que esse planejamento fosse realizado visando projeções futuras para uma grande cidade, não poderiam imaginar que o crescimento do tráfego de automóveis aumentaria desenfreadamente. Além disso, praticamente todas as metrópoles surgiram muito antes da invenção dos automóveis. Em Brasília-DF, esperava-se que no ano de 2000 o tráfego de veículos possuísse 100 mil automóveis, mas em 1996, o tráfego já ultrapassava 700 mil veículos (PAULO, 2010).

A Associação Nacional dos Detrans (AND) apontou em 2015 no Brasil que o número de condutores habilitados era de 60,7 milhões de motoristas em uma população de 204 milhões de habitantes Nacional (2015). Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), a população brasileira em julho de 2017 é 207 milhões de habitantes, e estima-se que em 10 anos, chegará a 220 milhões de habitantes IBGE (2017). Esse crescimento populacional implica que o número de cidadãos aptos a obter sua Carteira Nacional de Habilitação (CNH) aumentará consideravelmente ao longo dos anos.

Diante do que foi citado anteriormente, as vias de transporte urbano e rodoviário devem se adaptar de forma que atenda com segurança os impactos produzidos pelo crescimento constante do número de automóveis nas metrópoles brasileiras, causado por alguns fatores, como por exemplo, a facilidade de crédito financeiro nas concessionárias brasileiras para adquirir um veículo de passeio (ARAUJO; RODRIGO, 2016).

O crescimento citado anteriormente pode gerar vários transtornos no trânsito, no início da era automobilística, congestionamentos eram vistos apenas como situação desagradável aos motoristas. Entretanto, outros aspectos relevantes se fazem necessários discutir: (1) grandes centros possuem congestionamentos gigantescos, lugares estes que possuem um nível considerável de poluição atmosférica. (2) A população apresenta vulnerabilidades quanto à arrastões que eventualmente acontecem nos corredores produzidos pelo congestionamento de trânsito nas vias brasileiras (GRAEML; GRAEML, 1997).

Em 2009, o número de acidentes no trânsito do país aumentou de 19 por 100 mil habitantes para 23,4 por 100 mil habitantes até 2015, sendo assim o maior índice na América do Sul. No Brasil, o índice de óbitos oriundos de acidentes de trânsito é gigantesco. Segundo a Organização Mundial da Saúde (OMS), em 2015, mais de 43 mil pessoas perderam suas vidas em acidentes de trânsito, nas ruas e estradas OMS (2015). É imprescindível que vítimas de acidentes de trânsito sejam socorridas e atendidas o mais rápido possível, pois o primeiro atendimento à vítima realizado de forma rápida e correta

pode reduzir o risco de vida, e de futuras sequelas. E para que esse atendimento aconteça de forma rápida, faz-se necessário a utilização de VEs para tal resgate ([PRIME, 2017](#)).

Os VEs são veículos usados para fiscalizações, operação de trânsito, policiais, dentre outras atividade emergenciais. Estes veículos necessitam estar caracterizados de acordo com suas funções, equipados com alarme sonoro e iluminação intermitente, características estas obrigatórias segundo o Código de Trânsito Brasileiro (CTB, 2007, referente ao artigo 29 do inciso VII). Os VE têm prioridade e acesso livre no trânsito, em situação de emergência ([BRASIL, 2008](#))

Diante do problema apresentado, o intuito deste trabalho é contribuir para que os VE tenham segurança ao conduzirem uma ocorrência de emergência nas cidades brasileiras, possibilitando diminuição no risco de acidentes que possa prejudicar um paciente que esteja sendo conduzido dentro do VE, ou terceiros que se envolvam em um acidente.

Para contribuir com a mudança deste cenário atual, foi desenvolvido um protótipo que gerencia o fluxo de um semáforo, no qual sua função é tornar livre a passagem dos VE em vias onde haja congestionamento, independente do tempo e do fluxo dos veículos, para que não seja necessário reduzir a velocidade ou até mesmo parar em cruzamentos, conforme o CTB. O fluxo será liberado não apenas em casos de acidentes, mas em qualquer tipo de ocorrência que o VE esteja destinado a atender, visto que o bom funcionamento do serviço de emergência pode colaborar na solução de problemas da nossa sociedade.

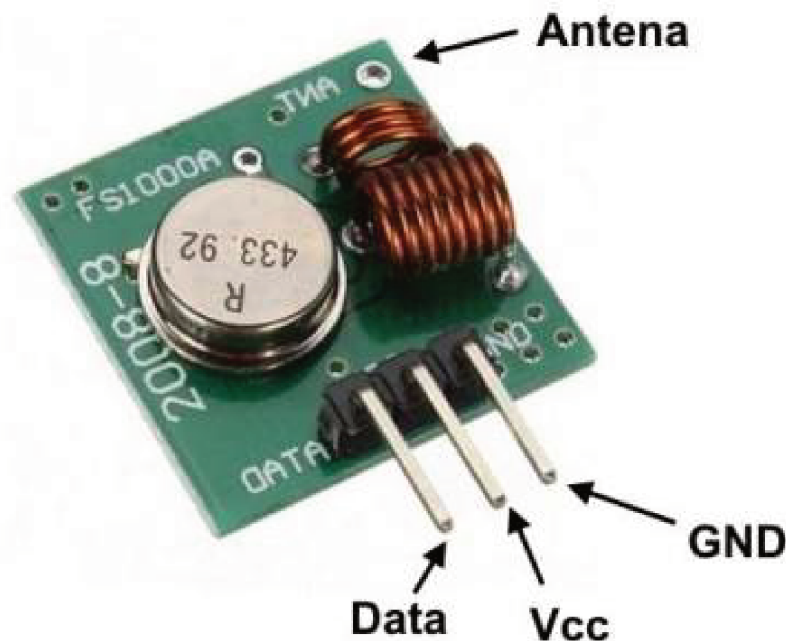
Tendo em vista o que é um VE, é possível que o protótipo citado anteriormente, seja instalado futuramente nos semáforos brasileiros, atuando como um receptor de um sinal de rádio frequência (RF) e dentro dos VE existirá um emissor RF. Dessa forma, tornando possível a comunicação entre eles. O objetivo deste trabalho é construir um protótipo que permita que um VE ao chegar em um cruzamento o mesmo realize a comunicação com o semáforo, solicitando a abertura do mesmo no sentido em que o VE trafega, de forma que não seja necessário reduzir a velocidade do VE ou parar no cruzamento em questão.

2 Trabalhos Relacionados

2.1 Comunicação de rádio frequência usando transmissor/receptor

(PAULA, 2015) propôs a construção de um robô que recebe instruções externas indicando como ele deve se portar mediante um evento, fazendo com que o mesmo se desloque até a bola e tenha como o objetivo o gol. E para que esses parâmetros fossem enviados de um meio externo ao robô fez-se o uso de uma comunicação sem fio. Em se tratando de Arduino, há várias maneiras de se obter uma comunicação sem fio. Uma forma de se realizar essa comunicação é através de módulos transmissor/receptor de RF. Paula (2015) optou por usar um dos módulos disponíveis no mercado utilizado para prototipagem, o módulo RF 433, que por sua vez possui um baixo custo e fácil montagem.

Transmissor: Este módulo possui 3 pinos, sendo 2 utilizados para a alimentação e 1 pino dedicado a receber os dados que serão transmitidos via RF. Além disso, possui um orifício no circuito onde é possível soldar uma antena para ampliar o alcance do sinal. Na Figura 1 é apresentado um modelo de transmissor de RF.

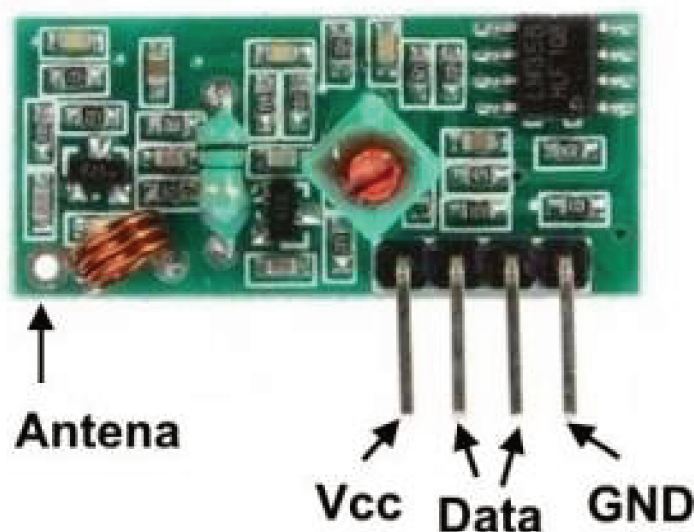


Fonte: www.blog.filipeflop.com (2015).

Figura 1 – O transmissor MX-FS-03V opera na frequência de 433,92MHz e pode ser alimentado com tensão entre 3,5Vcc e 12Vcc. E a distância de transmissão é entre 20m à 200m (depende da tensão), tendo como velocidade de transmissão de dados: 4 Kb/s. (PAULA, 2015)

Receptor: Este módulo possui 4 terminais, sendo 2 utilizados para a alimentação e 2 usados para a transmissão de dados, não havendo diferença entre estes. É possível utilizar apenas um terminal dos dois destinados a transmissão de dados, deixando o outro terminal livre. O módulo receptor também possui um orifício no circuito, onde é possível soldar uma antena que amplie a qualidade de recepção do sinal. A Figura 2 apresenta um modelo de receptor de RF.

Figura 2: Receptor de RF



Fonte: www.blog.filipeflop.com (2015).

Figura 2 – Este receptor observa a presença de dois pinos dedicados a transmissão de dados. Além disso, ele opera na faixa de frequência de 433,92 MHz e deve ser alimentado com 5 Vcc (PAULA, 2015).

A abordagem realizada por Paula (2015) torna possível observar com mais clareza sobre a comunicação entre um VE e um semáforo, uma vez que os sinais dos módulos de RF citados anteriormente são alimentados por uma tensão, e a distância de conexão entre um emissor e receptor depende do quanto de tensão é alimentado; onde é possível um alcance até 200 metros de distância entre um emissor e um receptor. Esta abordagem é útil para atender um dos objetivos desse trabalho.

2.2 O sensor de presença para os pedestres em semáforos

(CARDOSO, 2012) realizou um estudo da quantidade de pessoas que fazem o uso do sistema de diminuição de tempo do fluxo de veículos de passeio, onde os semáforos são equipados com o botão permitindo a travessia dos pedestres todos os dias e em diversas regiões do Brasil. Com o objetivo de melhorar esse processo, Cardoso (2012) desenvolveu dispositivo que substitui o atual método, onde pode ser visto na Figura 3.

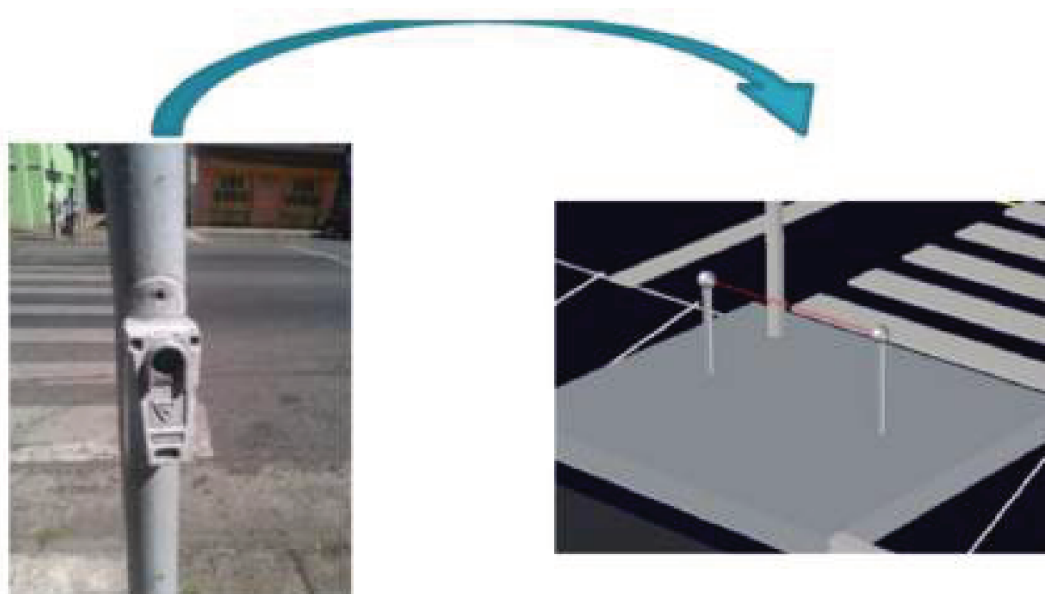


Figura 3 – A imagem do lado esquerdo representa o botão que pode ser acionado pelo pedestre atualmente. E a imagem do lado direito apresenta um infra-vermelho, onde após o pedestre cruzar o feixe, causa interrupção do receptor de luz infra-vermelho, simulando o acionamento do botão (CARDOSO, 2012).

A proposta do trabalho de Cardoso (2012) é que o sensor identifique pessoas (pedestres) através do sensor infra-vermelho no momento em que o mesmo se posicione próximo a faixa de pedestre, emita um beep (sinal audível) para o pedestre no momento em que o sistema o reconheça, auxiliando pessoas com deficiência visual; e para aquelas pessoas que possuem alguma deficiência auditiva, terão indicadores de LED (*Light Emitting Diode*), o qual apresenta a cor vermelha quando não é possível a travessia do pedestre, e a cor verde quando a passagem do pedestre é livre, ou seja, permitida.

A Figura 4 apresenta uma visão ampla da proposta de Cardoso (2012), onde pode-se observar a altura dos postes com o sensor infra-vermelho, possibilitando a identificação de pessoas com baixa estatura (e.g. crianças ou cadeirantes) e a distância entre o emissor e o receptor. Apesar das dificuldades em encontrar documentações sobre a linguagem de programação Assembly (linguagem de montagem) para suprir os desafios propostos pelo sistema, os resultados esperados foram alcançados com êxito, tornando possível a implantação do sistema em outros cenários de trânsito além do local de teste.

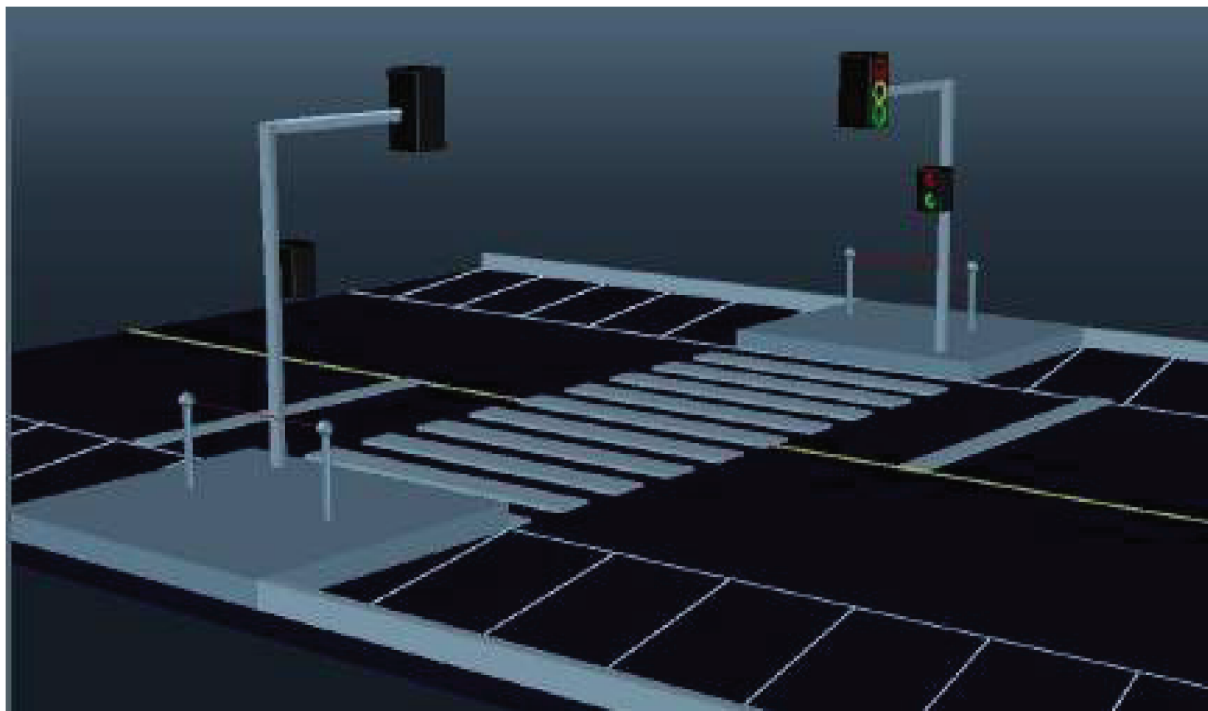


Figura 4 – Visão ampla do sistema de presença (CARDOSO, 2012).

O método de detecção de presença (ou objetos) usados por Cardoso (2012) é de grande relevância para o desenvolvimento deste trabalho, pois com esta abordagem é possível contar o fluxo de veículos nas vias de trânsito. Ademais, o sensor de presença pode ser usado na mesma plataforma de desenvolvimento (ARDUÍNO) proposto no trabalho.

2.3 Aquisição de dados usando os sensores de baixo custo

Conforme GALANTE e GARCIA (2014), o mercado tecnológico tem se voltado cada vez mais em solucionar problemas do dia a dia, suprimindo necessidades do campo agrícola, mostrando novas propostas de qualidade. Entretanto, surgem alguns obstáculos ao se tratar de negócios do campo, o primeiro pode ser o custo elevado financeiramente, e o segundo conhecimentos avançados dos usuários. GALANTE e GARCIA (2014) propõe um sistema de baixo custo para coletar temperaturas gerando mapa térmico (Figura 12).

As informações são produzidas mediante o sistema de aquisição de dados usando sensores de baixo custo, independente de marca, no caso os autores usaram a plataforma ARDUÍNO. A localização necessária para construir o mapa térmico foi obtida através de um GPS (*Global Position System*) acoplado ao ARDUINO, e a temperatura foi obtida por meio de um sensor de temperatura simples (*LM35*). Assim como foi realizado o uso do sensor de temperatura, podem ser usados outros sensores, como por exemplo: sensor de umidade, sensor de luminosidade, entre outros. Todos os dados obtidos pelos sensores são gravadas no cartão de memória instalado no ARDUINO, gerando um arquivo

DATALOG.TXT contendo as informações de localização e os dados de temperatura que são gerados de 1 em 1 segundo, e posteriormente importados no software Web Fazenda, que é responsável por gerar o mapa térmico. A abordagem usada por GALANTE e GARCIA (2014) com GPS e como as informações foram coletadas, são essenciais para que possamos compreender o mapeamento de rotas, que os VE possivelmente trafegarão nas vias.

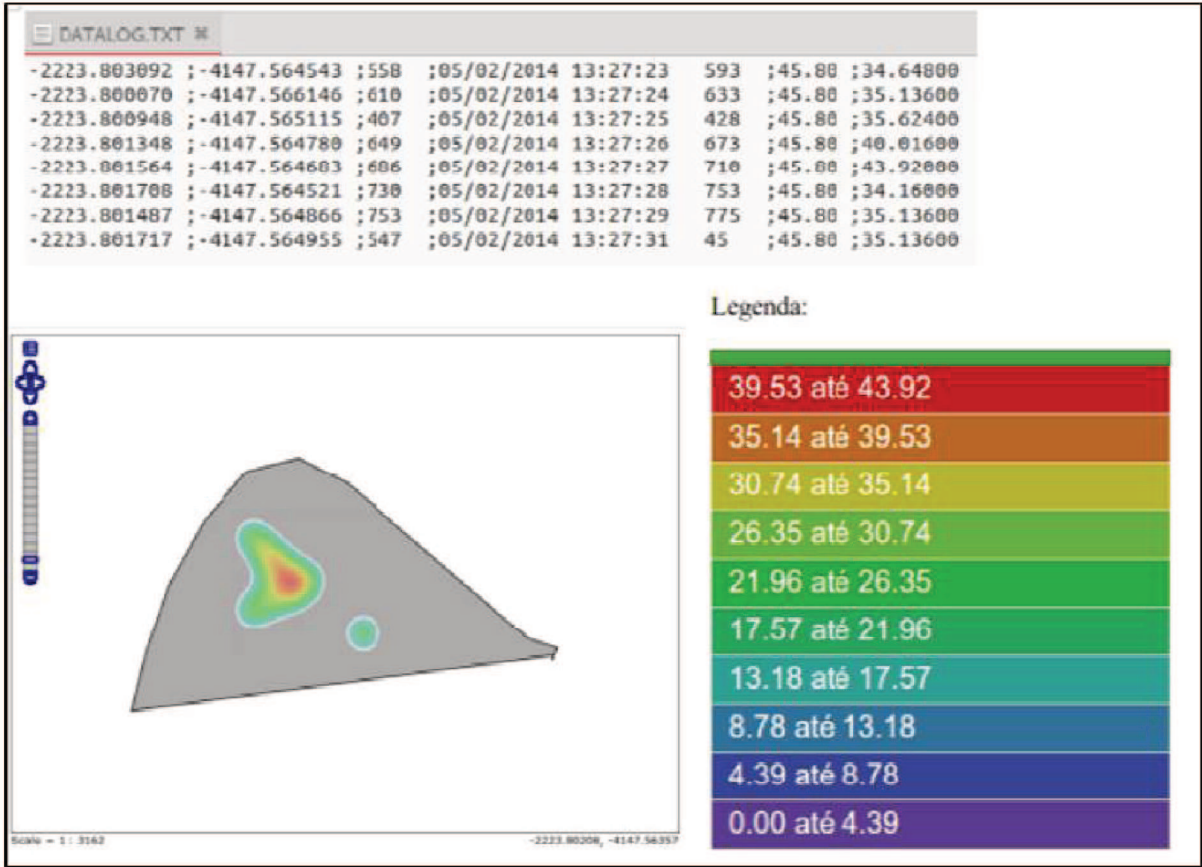


Figura 5 – Mapa gerado pelo sistema de aquisição (GALANTE; GARCIA, 2014).

3 Metodologia

O desenvolvimento deste trabalho é a construção de um Semáforo Inteligente para Veículos de Emergência. Assim, a metodologia proposta foi dividida em três etapas.

3.1 Construção da Maquete

Primeiramente deu-se início à etapa de construção de uma maquete que simulasse como seria o funcionamento do sistema do semáforo inteligente, nessa etapa também foram adquiridos os sensores necessários para o funcionamento do semáforo real seguindo as especificações do estudo proposto conforme apresenta a Figura 6.

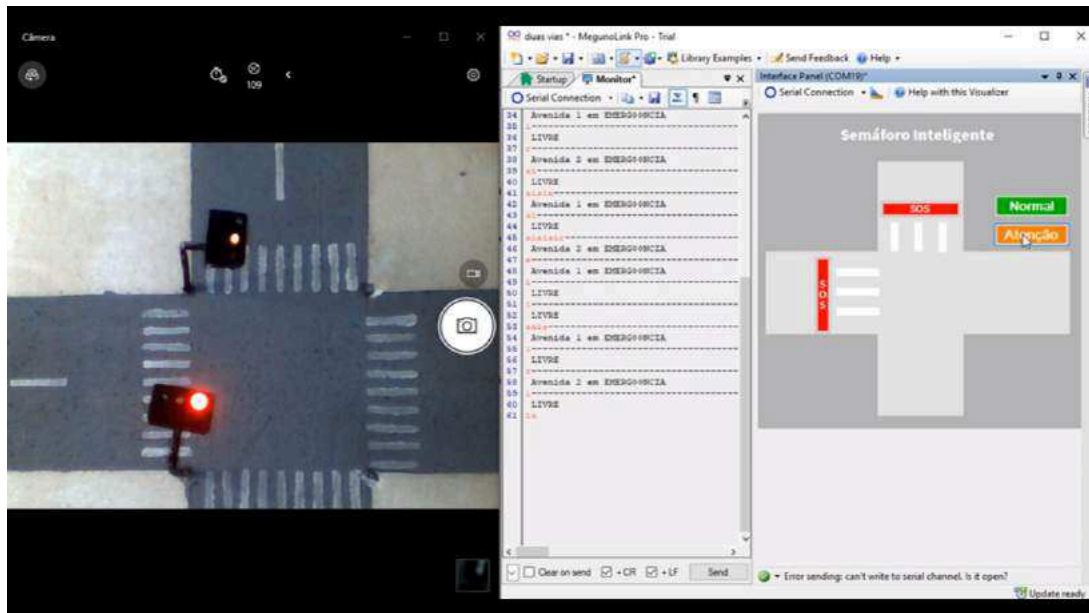


Figura 6 – Maquete construída e desenvolvida para simular um cruzamento com dois semáforos inteligentes em funcionamento real com o Sistema Inteligente.

3.2 Implantação de Sensores

Foi implantado no sistema já em funcionamento na maquete da segunda etapa, o sensor de Rádio Frequência 433, que permite a comunicação remota entre dois dispositivos, tornando possível a comunicação entre um veículo de emergência e um semáforo equipado com o sistema inteligente conforme a Figura 7.

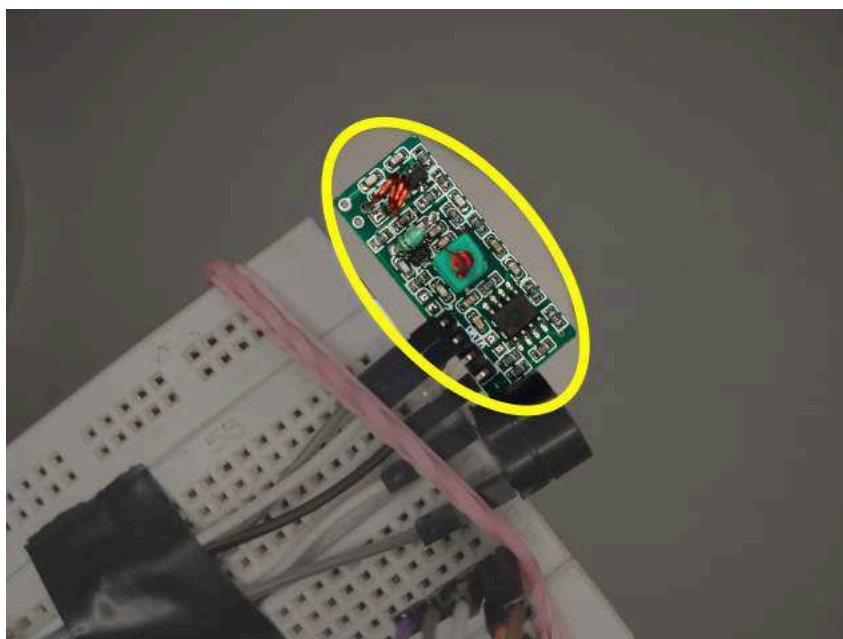


Figura 7 – Adicionado o sensor receptor na Maquete desenvolvida da segunda etapa.

3.3 Construção do Semáforo Real

Nesta etapa foi construída um semáforo em escala real, onde foram acrescentados módulos relês para controlar o funcionamento das lâmpadas do semáforo, que por sua vez funcionam numa tensão de (100V 220V), sem a integração desses relês seria impossível o micro controlador Arduino alimentar as lâmpadas do semáforo, uma vez que o Arduino não fornece a tensão necessária pra acender uma lâmpada (Figura 8).

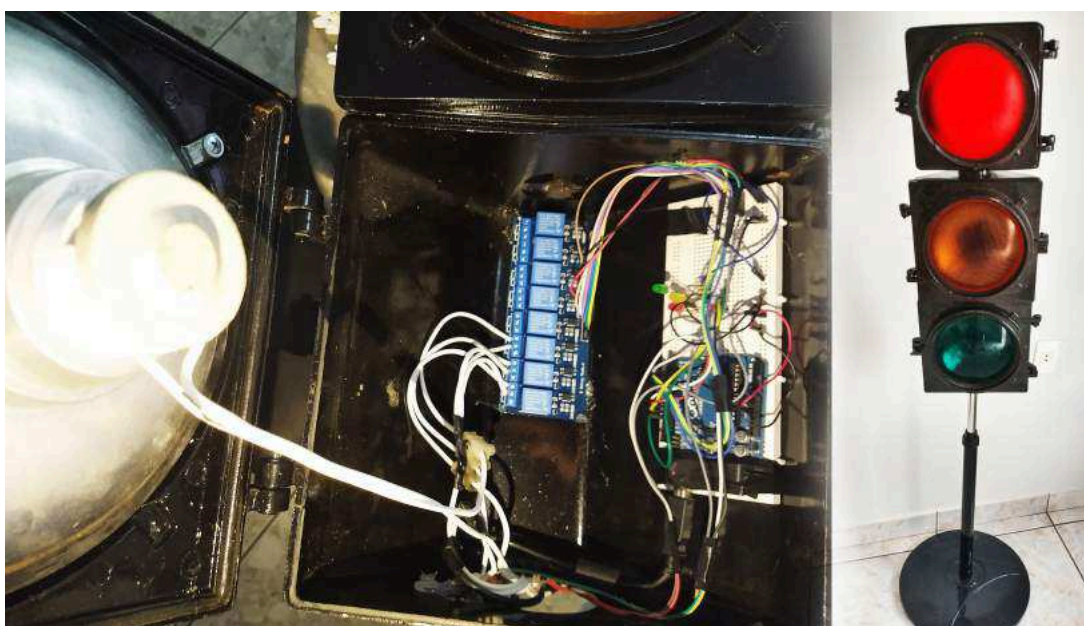


Figura 8 – Semáforo Inteligente.

3.4 Hardware

Os micro controladores por si só, não dispõem uma plataforma de desenvolvimento, então faz-se necessário o uso de outros componentes como: reguladores de tensão, resistores, diodos, etc. Criando assim de fato um Hardware para uma aplicação específica. No entanto, o propósito deste estudo é elaborar uma solução de baixo custo, construir um micro controlador componente a componente elevaria o custo da solução em uma razão considerável.

Tendo em vista este obstáculo, optou-se pela plataforma de prototipagem Arduino para a construção da solução proposta devido ao custo financeiro, uma vez que a plataforma proposta oferece todos os componentes citados no parágrafo anterior, integrados em uma única placa de circuitos impressos. O ARDUINO oferece uma vasta compatibilidade com os mais variados tipos de sensores digitais ou analógicos existentes no mercado, de forma simples e com um custo baixo, oferecendo assim, todas as ferramentas necessárias para o desenvolvimento da solução proposta. Estes fatores foram de suma importância na escolha desta plataforma para o desenvolvimento deste estudo.

3.5 Plataforma ARDUINO

A plataforma de prototipagem ARDUINO oferece inúmeras opções de soluções para um mesmo tema, assim sendo, proporcionando uma oportunidade de escolha da qualidade e investimento. Desta forma, vejamos algumas de suas características.

- Alimentação externa com pino Jack com positivo no centro, ou alimentação via USB (5V) e conta com um regulador de tensão de entrada, possibilitando o uso de fontes de alimentação que possuam tensão entre 6 a 20 volts, porém se a tensão de entrada for inferior a 7V, o funcionamento da placa pode ficar instável dependendo do sensor usado, e se alimentado por uma tensão superior a 12V a placa pode sofrer superaquecimento causando danos na placa e nos sensores conectados à mesma.
- Saída de alimentação de 3,3 e 5 volts para ligar outros componentes ao ARDUINO sem ter necessidade de usar outra fonte de alimentação, e existe uma saída VIN, que fornece a mesma tensão da fonte de alimentação externa através de um pino Jack.
- Conectores em todas as portas de entrada e saída do micro controlador facilitando a integração com outros componentes, os sensores usados no dispositivo.
- A plataforma possui um ambiente de desenvolvimento de algoritmos bastante considerável e upload através da porta USB de qualquer computador para a placa.

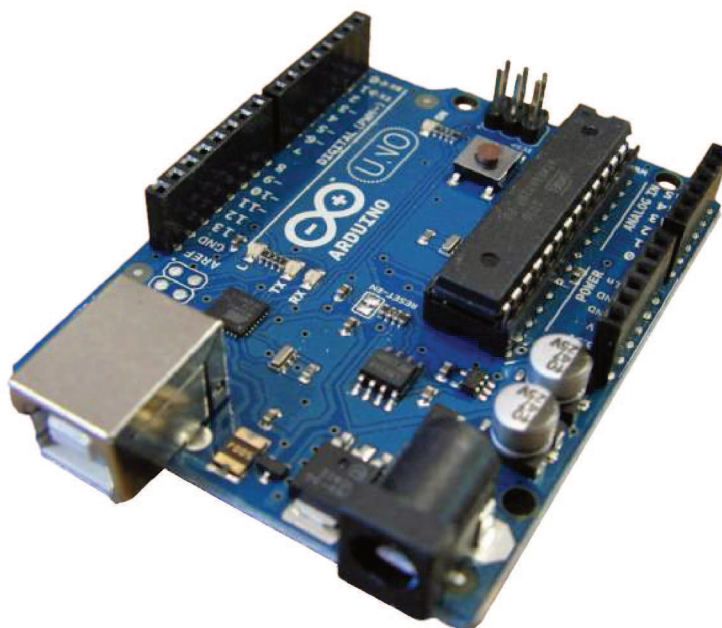


Figura 9 – ARDUINO UNO ([ARDUINO](#), 2018).

3.6 Sensor de Rádio Frequência RF433

O módulo RF433MHz Transmissor/Receptor (Figura 10) é uma opção com custo baixo e bastante eficiente quando a velocidade de transmissão de dados não é um fator preocupante. O módulo oferece uma comunicação Wireless, possibilitando o trânsito de dados entre placas ARDUINO de forma remota, sem a necessidade do uso de fios.

Esse módulo trabalha de forma unidirecional, ou seja, dados são enviados pelo transmissor, e recebidos pelo receptor, os dois sensores trabalham em par, para uma possível comunicação bidirecional seria necessários 2 pares do módulo RF-433MHz.

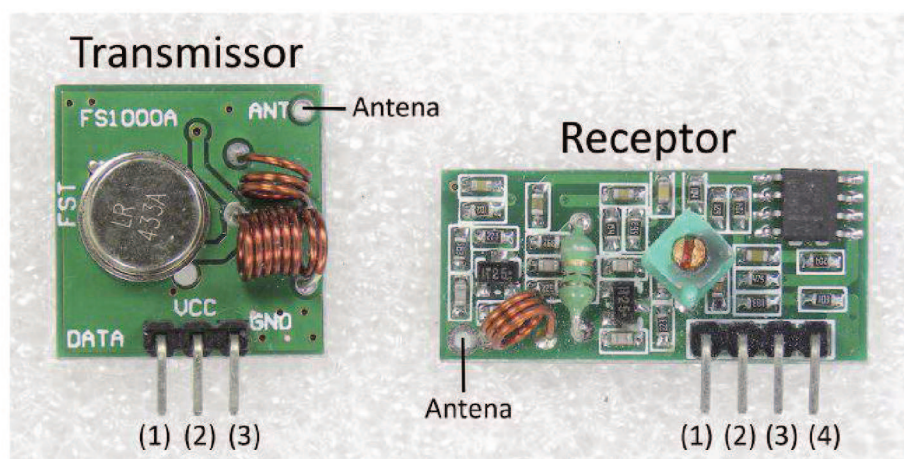


Figura 10 – Módulo RF 433MHz ([433MHZ](#), 2018).

3.6.1 Pinagem dos Módulos TX/RX de 433 MHz

Módulo Transmissor

- 1 - DATA – pino de transmissão de dados
- 2 - VCC – pino de alimentação : 3 a 12 V
- 3 - GND – terra

Módulo Receptor

- 1 - VCC – pino de alimentação : 5V (Somente)
- 2 - Data – pino de recepção de dados
- 3 - Data – idem – conectado ao pino 2
- 4 - GND – terra

3.6.2 Especificações dos Módulos TX/RX de 433 MHz

Módulo Transmissor 433 MHz

Tensão de operação: 3 a 12 V

Frequência de operação: 433,92 MHz

Corrente de operação: 20 a 28 mA

Potência de saída : 10mW

Taxa de transferência < 4Kbps

Alcance da transmissão < 100 m (com antena, sem obstáculos)

Antena externa: fio com 17,3 cm

Módulo Receptor 433 MHz

Tensão de operação: 5 V (somente)

Frequência de operação: 433,92 MHz

Corrente de operação: 4 mA

Sensibilidade de recepção: -105DB

Antena externa: fio com 17,3 cm

3.6.3 Antena

Os módulos funcionam mesmo sem nenhuma antena acoplada no sensor, porém a distância de comunicação entre um sensor e outro não pode ser grande, em nossos teste essa distância foi no máximo 5 metros. Portanto, a fim de um bom funcionamento do

módulo, se faz necessário soldar uma antena no transmissor e no receptor, essa antena pode ser um arame rígido 17,3 cm para dois sensores comunicarem adequadamente.

3.6.4 Alcance

Existem dois fatores cruciais acerca do alcance de comunicação entre os sensores: antena e tensão de alimentação dos sensores. Como já foi dito antes, os sensores funcionam sem antena, porém com uma pequena distância, com o uso adequado das antenas e os dois sensores alimentados com tensão igual a 5V a distância de comunicação pode ultrapassar os 100 metros. Essa distância pode ser ampliada alimentando somente o transmissor com uma tensão variável de (5A a 12V), alimentando-o com 12V a distância de comunicação pode chegar a 200 metros sem obstáculos, segundo ([ELETROGATE, 2017](#)).

3.7 Módulo relê

O Módulo Relê (Figura 11) torna possível acionar energia de um dispositivo elétrico qualquer, onde a carga exigida seja no máximo 10A contínuo, como por exemplo: Uma lâmpada, ventilador, bomba d'água, etc. utilizando a plataforma ARDUINO ou qualquer outro micro controlador para definir como será o fluxo de acionamento desses relês.



Figura 11 – Módulo Relê ([RELé, 2018](#)).

3.7.1 Esquema de Ligação do Módulo Relê

Esse módulo funciona como um interruptor de luz, mas ele depende de um micro controlador que definirá quais circunstâncias o mesmo deverá ser acionado de acordo com algoritmo útil para o tema([ROBOCORE](#),).

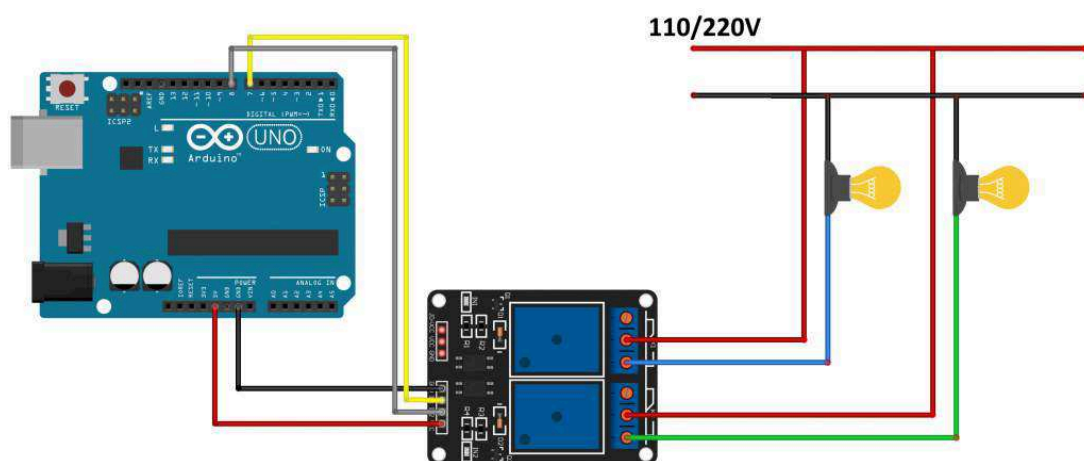


Figura 12 – Esquema de ligação do módulo relê. ([FLIPEFLOP](#), 2017)

4 Resultados

4.1 Contrução do semáforo

Através de uma doação do setor de obras da prefeitura municipal de Monte Carmelo-MG, foi adquirido 3 carcaças de semáforos com as lentes verde, amarelo e vermelho conforme apresenta a Figura 13, o que possibilitou ter o semáforo inteligente em tamanho real para realizar os testes necessários para desenvolvimento do "Semáforo Inteligente".



Figura 13 – Carcaça de um semáforo real.

Para que a maquete pudesse ser executado nesse semáforo foi instalado os módulos relês (Figura 14), onde então foram instaladas as lâmpadas cujo a tensão varia de (127V 220V), o que permitiu que o ARDUINO pudesse controlar essas lâmpadas.



Figura 14 – Semáforo equipado com lâmpadas e relês.

4.2 Teste de 5 metros

Foi realizado testes de distância de comunicação entre o transmissor e o receptor. O primeiro teste foi feito em um ambiente bastante poluído sonoramente, falando-se de Rádio frequência, onde haviam vários carros, celulares, antenas e diversos outros fatores que poderiam influenciar na comunicação entre o veículo e o semáforo. O transmissor alimentado via USB cuja tensão fornecida é de 5V foi posicionado dentro do carro, e o receptor alimentado via USB foi colocado numa distância de 5 metros (Figuras 15 16).

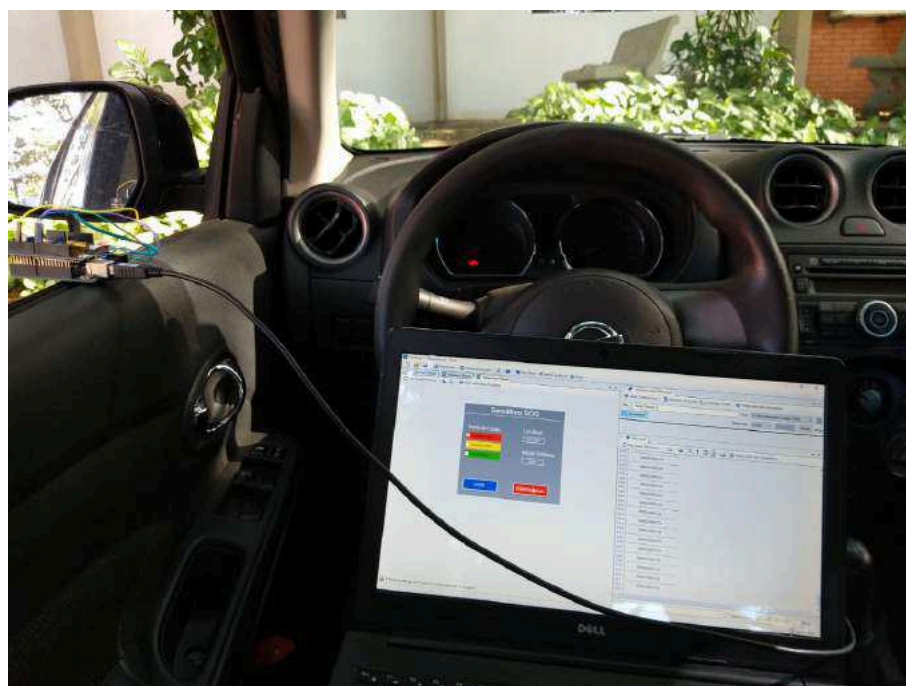


Figura 15 – Teste de 5 metros, transmissor sem antena, de dentro do carro.

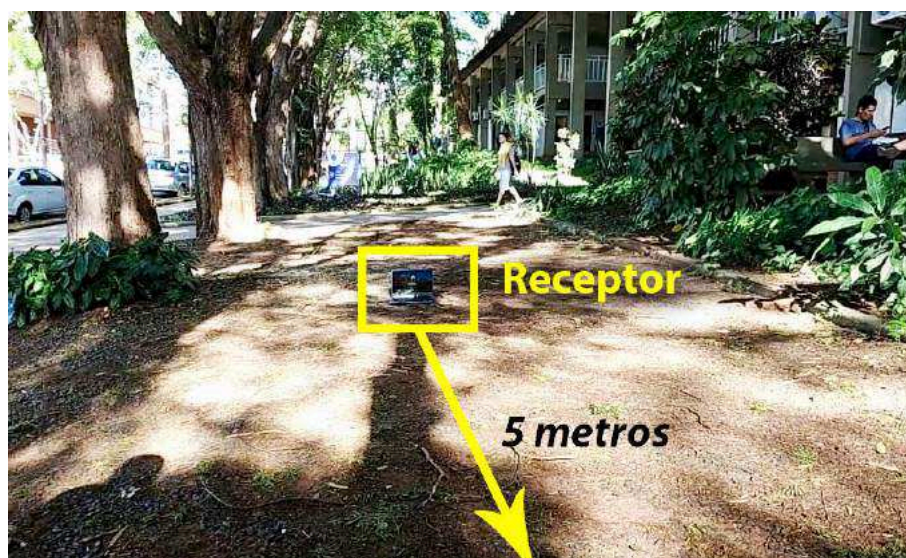


Figura 16 – Teste de 5 metros, receptor (Semáforo) sem antena, ambiente aberto.

Ao se distanciar além dos 5 metros, foi constatado que a conexão entre o semáforo inteligente e o veículo de emergência ficou instável ou até mesmo perdia alguma vezes a sua conexão, tendo que se aproximar à linha dos 5 metros novamente nos testes realizados.

4.3 Teste de 10 Metros

Em outro cenário real, agora desta vez, em uma rua remota onde não haviam tantos carros, e também com poucas casas, conseqüentemente não havendo tanta interferência nos sinais de rádio frequência, foi realizado o teste real com o Semáforo (Receptor) alimentado via USB (5V), e o veículo (Transmissor) também alimentado via USB.



Figura 17 – Teste de 10 metros, com antena, ambiente aberto.

Pôde perceber nesse experimento, que o ambiente aberto em que os ensaios foram realizados, afetava diretamente na conexão entre os sensores de Rádio frequência, pois nas mesmas condições e configurações de hardware, houve um ganho de distância na comunicação entre os dois dispositivos. Entretanto, nessa distância de 10 metros ao se aproximar de um veículo os dispositivos perdiam a conexão no mesmo instante.

4.4 Teste Final

Por fim, foi concluído toda a montagem do sistema inteligente, conforme mostra a Figura 18; então, colocando dentro do semáforo o dispositivo que controla o fluxo do semáforo e ao mesmo tempo faz a comunicação com o veículo de emergência.



Figura 18 – Semáforo com teste de todas a lâmpadas.

Foi testado 3 tensões diferentes para alimentação do transmissor respectivamente (5V, 9V, 12V), buscando ter uma maior distância entre um sensor e o outro conforme vimos na seção 3.6.4. Contudo, independente da tensão em que o transmissor foi alimentado, não foi possível obter conexão com semáforo em uma distância superior a 5 metros.

4.4.1 Início do sistema

Assim que o semáforo é ligado à energia, ele acende todas a luzes, e permanece até que um sinal de um veículo inteligente seja identificado em sua rede (Figura 19).

4.4.2 Trânsito Livre

Assim que o transmissor é ligado e começa a gerar sinal de Rádio Frequência, quando esse sinal alcança o receptor, o sistema do semáforo inicia em um fluxo normal de trânsito (Verde, Amarelo e Vermelho), e continuará nesse laço até que seja enviado um pedido de emergência pelo VE (Figuras 20, 21, 22, 23).

4.4.3 Trânsito em Emergência

Para que o semáforo entre em estado de emergência, o transmissor deve acionar o botão de emergência, enviando pulsos de sinais com a opção de trânsito desejada, no caso de estar pedindo para que o semáforo entre em estado de emergência o transmissor enviará o número 1 (Um) e para voltar ao normal, o valor 0 (Zero) (Figura 24).



Figura 19 – Inicialização do semáforo inteligente.

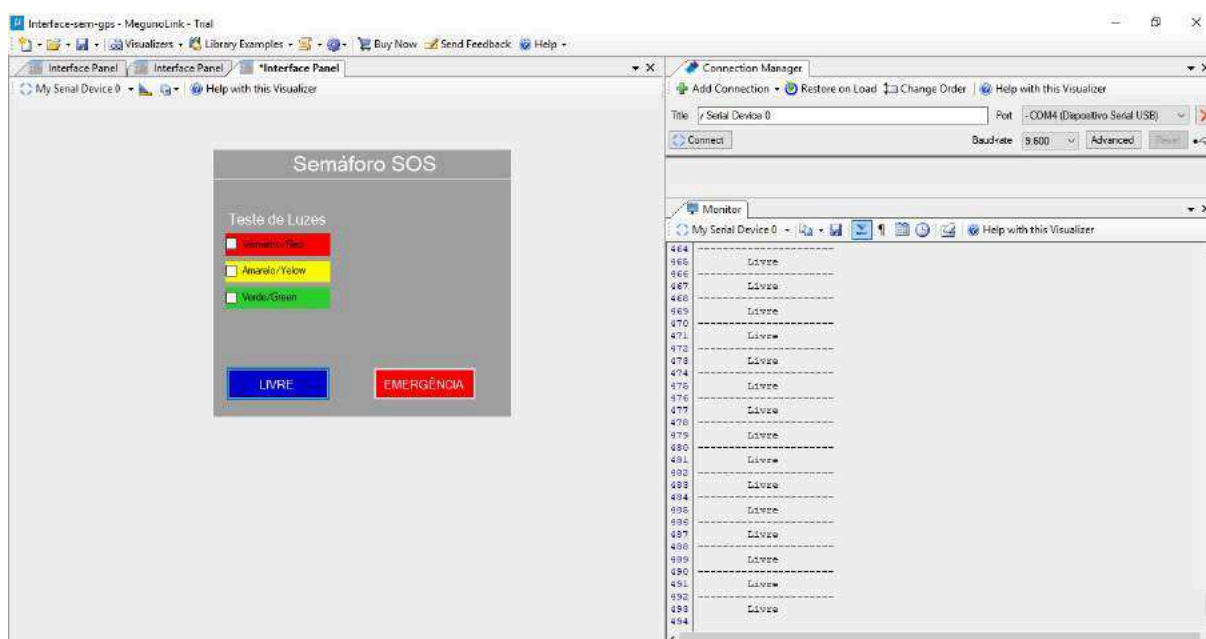


Figura 20 – Transmissor emitindo sinal de trânsito livre.

Quando o semáforo recebe o pedido de emergência, ele coloca o sinal em amarelo como forma de "Atenção"(Figura 25). Após ter sinalizado em forma de atenção com o sinal amarelo, o sistema abre a via em que foi solicitado a emergência, ficando com o sinal verde, garantindo a passagem do veículo de emergência pelo cruzamento com segurança (Figura 26). Por fim, O semáforo continuará verde enquanto o veículo estiver em seu raio de cobertura, então não tendo mais solicitação ativa, o semáforo volta ao fluxo normal.

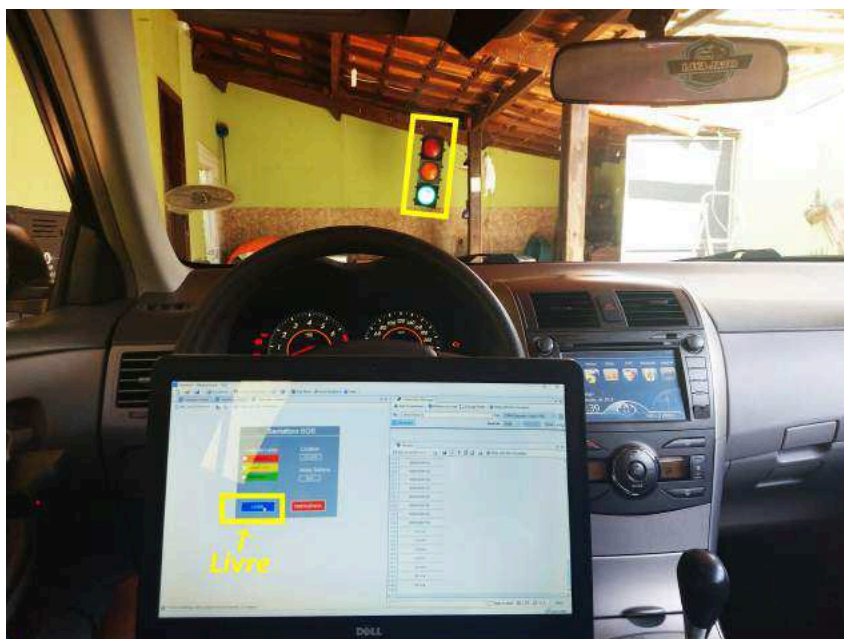


Figura 21 – Sinal verde durante fluxo normal.

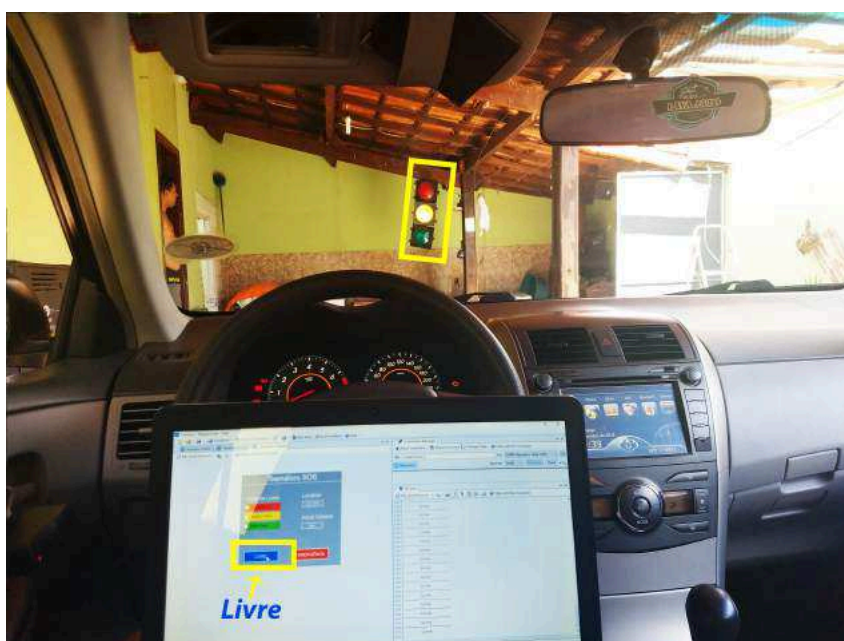


Figura 22 – Sinal Amarelo durante fluxo normal.

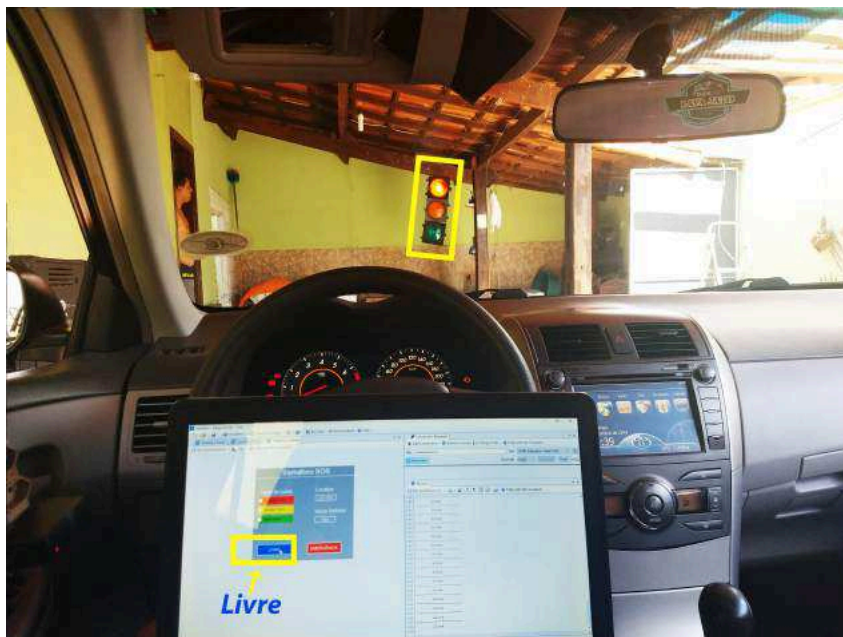


Figura 23 – Sinal vermelho durante fluxo normal.

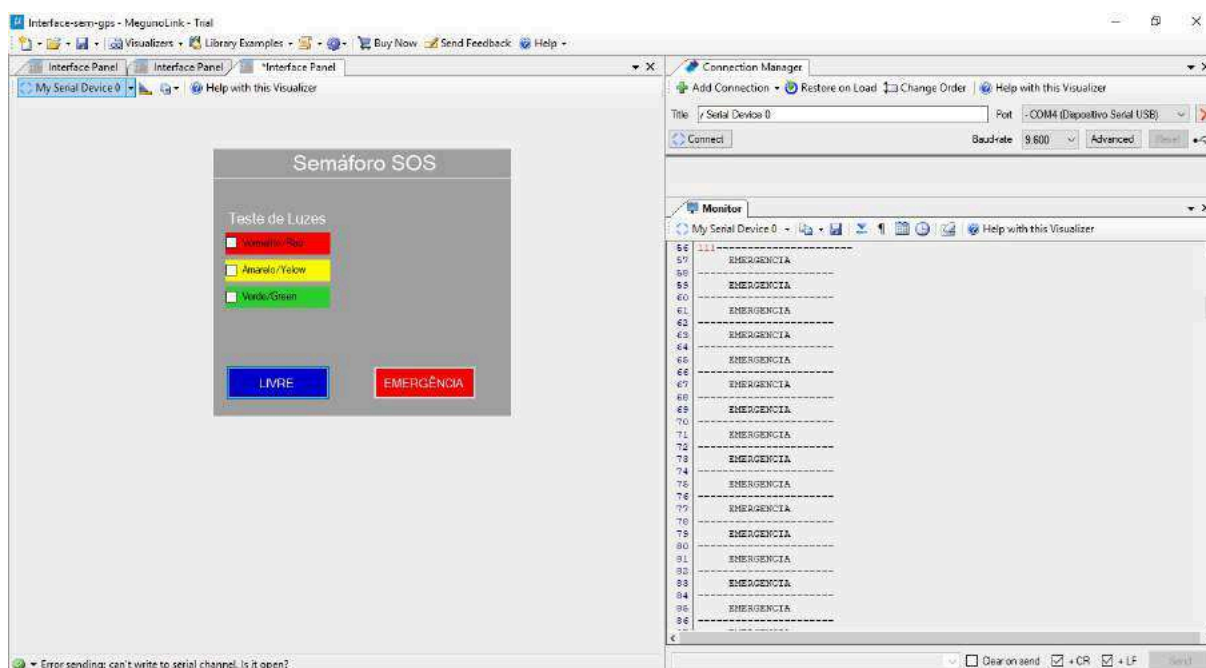


Figura 24 – Transmissor enviando requisição de abertura do sinal através da emergência.



Figura 25 – Semáforo em estado de atenção.



Figura 26 – Semáforo em estado livre.

Conclusão

O objetivo deste trabalho foi desenvolver um dispositivo baseado no ARDUÍNO, que gerencia a passagem dos VE por cruzamentos, onde há semáforos através de um sistema de comunicação entre o semáforo e o VE, tornando o atendimento à ocorrência ágil e segura. Assim, com este estudo possibilitou que testes reais acontecessem em vias urbanas. Foi uma fonte de aprendizado acerca de um mundo infinito que é a plataforma ARDUINO, com suas incontáveis possibilidades de soluções de problemas do cotidiano.

Com a solução proposta o usuário do sistema tem o poder de remotamente anteciper a abertura de um semáforo, apenas apertando um botão. Todo sistema pode ser adaptado conforme a necessidade do problema seja necessário o sentido de via, ou tempo de transição de um sinal para o outro. O fato de a plataforma escolhida ser de hardware aberto tanto em hardware quanto em software, faz com que os preços dos ARDUINOS e periféricos sejam baratos, tornando acessível à comunidade como um todo.

Os resultados através desse trabalho foram satisfatórios, uma vez que tudo partiu de uma maquete e chegou em um semáforo de escala real. Espera-se que este trabalho contribua com a comunidade acadêmica e comunidade social, visto que o tema que nos propusemos resolver afeta a todos, uma vez que o trânsito faz parte da vida de quase todos os cidadãos. Como ponto negativo podemos ressaltar a curta distância na comunicação entre os dois sensores, sendo um fator primordial, pois uma longa distância de comunicação é crucial para que aconteça todo trâmite de gerenciamento do trânsito com segurança.

Para trabalhos futuros, em relação a abertura do semáforo para os veículos de emergência, pretende-se implantar um GPS para que se obtenha de qual direção o VE está se aproximando e assim o Semáforo Inteligente possa tomar uma decisão segura de qual via deve ser liberada para a passagem dos Veículos de Emergência.

Referências

433MHZ, M. R. *Guia básico dos Módulos TX / RX – RF 433MHZ*. 2018. Disponível em: <<http://blog.eletrogate.com/guia-basico-dos-modulos-tx-rx-rf-433mhz/>>. Acesso em: 21 Dez. 2018. Citado 2 vezes nas páginas 5 e 19.

ARAUJO, H. dos S.; RODRIGO, J. Simulação de abertura em semáforos para veículos de emergência. 2016. Citado na página 9.

ARDUINO. *Arduino*. 2018. Disponível em: <<https://pt.wikipedia.org/wiki/Arduino>>. Acesso em: 21 Dez. 2018. Citado 2 vezes nas páginas 5 e 19.

BRASIL, C. d. T. B. *Código de trânsito brasileiro: instituído pela Lei nº 9.503*. [S.l.], 2008. Citado na página 10.

CARDOSO, H. S. M. Sensor de presença de pedestres em semáforos das comerciais do plano piloto. 2012. Citado 3 vezes nas páginas 5, 13 e 14.

ELETROGATE. *Guia básico dos Módulos TX / RX*. 2017. Disponível em: <<http://blog.eletrogate.com/guia-basico-dos-modulos-tx-rx-rf-433mhz/>>. Acesso em: 29 Nov. 2018. Citado na página 21.

FLIPEFLOP. *Controlando lâmpadas com Módulo Relé Arduino*. 2017. Disponível em: <<https://www.flipeflop.com/blog/controle-modulo-rele-arduino/>>. Acesso em: 8 Dez. 2018. Citado 2 vezes nas páginas 5 e 22.

GALANTE, A. C.; GARCIA, R. F. Sistema de aquisição de dados de sensores de baixo custo baseado no arduíno. In: *Congresso Brasileiro de Agricultura de Precisão*. [S.l.: s.n.], 2014. v. 6. Citado 3 vezes nas páginas 5, 14 e 15.

GRAEML, A. R.; GRAEML, F. R. A automação do tráfego de veículos (tecnologia aeroespacial/militar x tecnologia de chão de fábrica). *Gramado: Anais do XVIII ENEGEP, Encontro Nacional de Engenharia de Produção*, 1997. Citado na página 9.

IBGE. *População brasileira*. 2017. Disponível em: <<http://www.ibge.gov.br/apps/populacao/projecao>>. Acesso em: 09 Jul. 2017. Citado na página 9.

NACIONAL, R. *Você sabe quantos milhões de motoristas tem no Brasil?* 2015. Disponível em: <<http://www.radarnacional.com.br/voce-sabe-quantos-milhoes-de-motoristas-tem-no-brasil-confira-perfis-por-estado>>. Acesso em: 09 Jul. 2017. Citado na página 9.

OMS. *Brasil é o país com maior número de mortes de trânsito por habitante da América do Sul*. 2015. Disponível em: <<https://nacoesunidas.org/oms-brasil-e-o-pais-com-maior-numero-de-mortes-de-transito-por-habitante-da-america-do-sul>>. Acesso em: 10 Jul. 2017. Citado na página 9.

PAULA, C. J. H. d. Projeto e construção de um robô autônomo com visão computacional. Edson Simões dos Santos, 2015. Citado 3 vezes nas páginas 5, 11 e 12.

PAULO, R. F. O Desenvolvimento Industrial E O Crescimento Populacional Como Fatores Geradores Do Impacto. p. 173–189, 2010. Citado na página 9.

PRIME. *A importância dos Primeiros Socorros*. 2017. Disponível em: <<https://www.primecursos.com.br/blog/prime-cursos/a-importancia-dos-primeiros-socorros/>>. Acesso em: 10 Jul. 2017. Citado na página 10.

RELÉ, M. *Módulo Relé 4 canais 5V*. 2018. Disponível em: <<https://bit.ly/2Csta5b>>. Acesso em: 21 Dez. 2018. Citado 2 vezes nas páginas 5 e 21.

ROBOCORE. *Módulo Relé Arduino, year = 2018, url = <https://www.robocore.net/tutorials/modulo-rele-arduino.html>, urlaccessdate = 21 Nov. 2018*. Citado na página 22.

Apêndices

```
transmitter$
#include <VirtualWire.h>
//Programa : Comunicacao com Modulo RF 433 - Transmissor
//Autor : Samuel Izaias Alves
//Auxiliares

int cont1,cont0;
char serialOn,cmd,cmdIn,serialAux,r,y,g;

//Define pinos Led e Botao
int vcc=8;
int key=9;
const int ledPin = 13;
const int pino_botao = A0;

int valor_botao;

//Envia a mensagem do estado do botão
char Valor_CharMsg[4];

//Armazena estado led = ligar/desligar
int estado = 0;
```

Figura 27 – Declaração de variáveis do TRANSMISSOR.

```
void setup()
{
    Serial.begin(9600);
    pinMode(vcc,OUTPUT);
    pinMode(key,INPUT);
    pinMode(pino_botao,INPUT); //Pino ligado ao pino DATA do transmissor RF
    vw_set_tx_pin(3);

    vw_setup(1000); //Velocidade de comunicacao (bits por segundo)
    digitalWrite(vcc,HIGH);
    Serial.println("_____");
    Serial.println(" Semaforo Inteligente - LIGADO");
    Serial.println("_____");

    //Zera os contadores
    cont1=0;
    cont0=0;
}
```

Figura 28 – Função de configuração.

```
void loop()
{
    cmd=Serial.read();
    if(cmd=='i')
        cmdIn='i';
    if(cmd=='l')
        cmdIn='l';

    //Verifica o estado do push-button
    valor_botao = digitalRead(key);
    serialOn = Serial.read(); // Recebe o que foi digitado no monitor serial
    itoa(valor_botao,Valor_CharMsg,10);

    if(serialOn=='1')
        serialAux='1';
    if(serialOn=='0')
        serialAux='0';

    if (valor_botao == 1 || serialAux=='1'){
        //Mensagem a ser exibida no monitor serial...

        Serial.println("-----");
        Serial.println("    EMERGENCIA    ");

        itoa(1,Valor_CharMsg,10);
        vw_send((uint8_t *)Valor_CharMsg, strlen(Valor_CharMsg));
        vw_wait_tx();
        delay(1000);
    }
}
```

Figura 29 – Função que executa o fluxo que ficará em Loop.

```
else{
    //Mensagem a ser exibida no monitor serial...
    Serial.println("-----");
    Serial.println("    Livre    ");

    //Proceso de conversão dos valores, e envio
    valor_botao = 0;
    itoa(valor_botao,Valor_CharMsg,10); // Converte a mensagem pra envio
    vw_send((uint8_t *)Valor_CharMsg, strlen(Valor_CharMsg));
    vw_wait_tx();
    delay(1000);
}
}
```

Figura 30 – Continuação da função Loop.

```
receiver
//Programa : Comunicacao com Modulo RF 433 - Receptor

#include <VirtualWire.h>

//Define pino led
int ledPin = 9;
int um=4,dois=5,tres=6; //Numero dos Relês
int lum=10, ldois=11, ltres=12; // Numero dos LEDs
int valor_recebido_RF, flagT=0;
char recebido_RF_char[4];
```

Figura 31 – Declaração de variáveis do RECEPTOR.

```
void setup()
{
    //Configuração dos relês
    pinMode(um,OUTPUT);
    pinMode(dois,OUTPUT);
    pinMode(tres,OUTPUT);
    alloff();

    //Configuração dos LED
    pinMode(lum,OUTPUT);
    pinMode(ldois,OUTPUT);
    pinMode(ltres,OUTPUT);

    //-----
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
    //Pino ligado ao pino DATA do receptor RF
    vw_set_rx_pin(3);
    //Velocidade de comunicacao (bits por segundo)
    vw_setup(1000);
    //Inicia a recepcao
    vw_rx_start();
    Serial.println("-----");
    Serial.println("Aguardando Sinal da Viatura .... ");
    Serial.println("-----\n");
    allOn();
}
```

Figura 32 – Função de configuração.

```

//-----
void loop()
{
    recebeSinal();
}
//-----
void recebeSinal(){
    uint8_t buf[VW_MAX_MESSAGE_LEN];
    uint8_t buflen = VW_MAX_MESSAGE_LEN;
    int flagE;

    if (vw_get_message(buf, &buflen))
    {
        int i;
        for (i = 0; i < buflen; i++){
            //Armazena os caracteres recebidos
            recebido_RF_char[i] = char(buf[i]);
        }
        recebido_RF_char[buflen] = '\0';

        //Converte o valor recebido para integer
        valor_recebido_RF = atoi(recebido_RF_char);
        // delay(100);
        Serial.println (recebido_RF_char);
        //Mostra no serial monitor o valor recebido
        //Altera o estado do led conforme o numero recebido
    }
}

```

Figura 33 – Função Loop.

```

if (valor_recebido_RF == 1){
    if(flagT==0){
        transicao();
        flagT++;
    }
    Serial.println("| Transito em EMERGENCIA!|");
    Serial.println("|_____|");

    emergencia();
}
// delay(100);
if (valor_recebido_RF == 0){
    flagT=0;

    Serial.println("| Transito LIVRE! |");
    Serial.println("|_____|");

    fluxoNormal();
}
}
/*else{
    Serial.println("| Sem Comunicacao |");
    Serial.println("|_____|");
}*/
}

```

Figura 34 – Função de acionamento dos relês.

```
void rum() {
    //relês
    digitalWrite (um, 0);
    digitalWrite (dois, 1);
    digitalWrite (tres, 1);
    //LED
    digitalWrite (lum, 1);
    digitalWrite (ldois, 0);
    digitalWrite (ltres, 0);
    delay(500);
}
void rdois() {
    digitalWrite (um, 1);
    digitalWrite (dois, 0);
    digitalWrite (tres, 1);
    //LED
    digitalWrite (lum, 0);
    digitalWrite (ldois, 1);
    digitalWrite (ltres, 0);
    delay(500);
}
void rtres() {
    digitalWrite (um, 1);
    digitalWrite (dois, 1);
    digitalWrite (tres, 0);
    //LED
    digitalWrite (lum, 0);
    digitalWrite (ldois, 0);
    digitalWrite (ltres, 1);
    delay(500);
}
```

Figura 35 – Continuação da função de acionamento dos relês.

```

//-----
void fluxoNormal() {
    rtres();
    rdois();
    rum();
}
//-----
```

Figura 36 – Fluxo do normal do semáforo.

```
//-----  
void transicao() {  
    digitalWrite(um, 1);  
    digitalWrite(dois, 0);  
    digitalWrite(tres, 1);  
    //LED  
    digitalWrite(lum, 0);  
    digitalWrite(ldois, 1);  
    digitalWrite(ltres, 0);  
    delay(2000);  
}  
//-----
```

Figura 37 – Função que faz a transição do sinais para emergência.

```
//-----  
void emergencia() {  
    digitalWrite(um, 1);  
    digitalWrite(dois, 1);  
    digitalWrite(tres, 0);  
    //LED  
    digitalWrite(lum, 0);  
    digitalWrite(ldois, 0);  
    digitalWrite(ltres, 1);  
    //Alarme  
    digitalWrite(ledPin, HIGH);  
    delay(30);  
    digitalWrite(ledPin, LOW);  
    delay(30);  
}  
//-----
```

Figura 38 – Função emergência.