



UNICEUB – CENTRO UNIVERSITÁRIO DE BRASÍLIA
FATECS - FACULDADE DE TECNOLOGIA
E CIÊNCIAS SOCIAIS APLICADAS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

Henrique Selvero Menezes Cardoso

**Sensor de presença de pedestres em semáforos das
comerciais do plano piloto**

Orientador(a): M.C. Maria Marony Sousa Farias

BRASÍLIA/DF

1º SEMESTRE DE 2012

HENRIQUE SELVERO MENEZES CARDOSO

**Sensor de presença de pedestres em semáforos das
comerciais do plano piloto**

Monografia apresentada ao Curso de Engenharia de Computação, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Orientador(a): M.C. Maria Marony Sousa Farias

BRASÍLIA/DF

1º SEMESTRE DE 2012

HENRIQUE SELVERO MENEZES CARDOSO

Sensor de presença de pedestres em semáforos das comerciais do plano piloto

Monografia apresentada ao Curso de Engenharia de Computação, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Orientador (a): M.C. Maria Marony Sousa
Farias Nascimento

Este trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação, e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas – FATECS.

Prof. Abiezer Amarilia Fernandes
Coordenador do Curso

Banca Examinadora:

Prof^a Maria Marony Sousa Farias, mestre em Engenharia Elétrica – UFPB – PB
Orientadora

Prof. Vera Duarte
UniCEUB

Prof. Msc Francisco Javier de Obaldia Díaz
UniCEUB

Prof Antonio Barbosa Junior
UniCEUB

*“Escolhe um trabalho de que gostes,
e não terás que trabalhar nem um dia na tua vida.”
(Confúcio)*

AGRADECIMENTOS

Agradeço primeiramente a Deus por tornar este sonho de vida realidade, por me iluminar e me guiar para conquista de meus objetivos pessoais e profissionais.

Agradeço a minha família Sylvio, Marta, Vera e Pongo por me ajudarem em toda minha vida e, em especial, nesses cinco anos de curso, para me tornar um ENGENHEIRO DE COMPUTAÇÃO.

Agradeço a todos os professores do UniCEUB e aos amigos e colegas de turma que me acompanharam nessa trajetória, em especial Rodrigo Monção e Willian Verri, que hoje estão finalizando suas respectivas monografias junto comigo e aos que ficaram pelo caminho mas, em breve, sentirão a mesma emoção que sinto neste momento.

Agradeço a minha professora orientadora Maria Marony por me ajudar nos momentos de dificuldade na confecção deste projeto e a todos os amigos que me ajudaram direta ou indiretamente para a realização desse sonho.

RESUMO

Este trabalho visa mostrar uma alternativa de solução que poderia substituir o botão para pedestres situado em sinais de trânsito, que, quando pressionados, têm o objetivo de diminuir o tempo de espera dos mesmos, fechando o trânsito de veículos em um menor espaço de tempo. Tal botão pode ser encontrado em vários pontos do DF, inclusive nas comerciais do plano piloto, foco principal deste trabalho. Para tal, será utilizado um sensor de presença, que, após a identificação do pedestre, irá enviar tal informação a um microcontrolador que irá possibilitar a passagem do mesmo no menor período de tempo possível. Além disso, um som será emitido, para que o pedestre tome conhecimento de que o sistema o identificou e está em funcionamento.

Palavras chave: microcontrolador, sensor, segurança, pedestres.

ABSTRACT

This paper aims to show an alternative solution that could replace the knob located at stoplights that, when pressed, aims to reduce the waiting time for pedestrians, closing vehicular traffic in a shorter space of time. This button can be found in commercial master plan, the main focus of this work. This will use a presence sensor, which, after identification of pedestrian will send this information to a microcontroller which will allow the passage of even the smallest possible time. In addition, a sound is output, so that the pedestrian aware that the system is operating.

Keywords: microcontroller, sensor, security, pedestrian.

SUMÁRIO

Lista de Figuras	VIII
Lista de Quadros	IX
Lista de Abreviaturas e Siglas	X
CAPÍTULO 1: INTRODUÇÃO	11
1.1 Motivação e Posicionamento	12
1.2 Objetivos Gerais	13
1.3 Objetivos Específicos	13
1.4 Justificativa e Relevância do Tema	14
1.5 Diagrama de Blocos	15
1.6 Escopo do Trabalho	16
1.7 Problema a Ser Resolvido	19
1.8 Metodologias	19
1.9 Estrutura da Monografia	19
CAPÍTULO 2: FUNDAMENTOS TEÓRICOS	21
2.1 Controles de Trânsito	21
2.2 Semáforo Veicular	21
2.3 Código de Trânsito Brasileiro	23
2.4 Termos da Temporização	24
CAPÍTULO 3: DESCRIÇÃO DO HARDWARE	25
3.1 Microcontrolador PIC 16F628A	25
3.2 Organização da Memória	27
3.3 Circuito Sonoro	28
3.4 KIT do Microcontrolador 16F628A	29
3.5 Regulador de Tensão 7805	31
3.6 Comunicação Serial	32

3.7	Pinagem do Conector Serial	32
3.8	Conversor de Nível Serial para TTL.....	33
3.9	Infravermelho.....	35
3.10	Tipos de Detecção e Aplicações	35
3.11	Circuito Infravermelho	38
3.12	LED.....	39
CAPÍTULO 4: TESTES E IMPLEMENTAÇÃO		41
4.1	Planejamento.....	41
4.2	Plano de Tráfego do Projeto	41
4.3	Algoritmo.....	43
4.4	Software	44
4.5	Montagem do Protótipo	50
4.6	Orçamento.....	57
Capítulo 5: CONCLUSÃO		58
5.1	Sugestões de Trabalhos Futuros	59
Referências Bibliográficas		60
Apêndice A.....		63

LISTA DE FIGURAS

Figura 1.1 – Proposta do projeto	12
Figura 1.2 – Fluxograma do software.....	16
Figura 1.3 – Estrutura do trabalho	17
Figura 1.4 – Distâncias do sensor.....	18
Figura 3.1 - Pinagem do PIC 16F628A	25
Figura 3.2- Diagrama de blocos do PIC 16F628A	26
Figura 3.3- Estruturação do PIC 16F628A.....	27
Figura 3.4 – Circuito de sonorização (buzzer)	28
Figura 3.5 – Layout da placa ACEPIC 18	30
Figura 3.6 – Fonte do Kit	31
Figura 3.7 – Descrição do circuito de acionamento dos LEDs	31
Figura 3.8 – Conector DB9 com sua pinagem enumerada.	32
Figura 3.9- MAX232.	33
Figura 3.10 – Pinagem do MAX232.	33
Figura 3.11 – Porta serial – MAX232 – Microcontrolador	35
Figura 3.12 - Detecção por reflexão.	36
Figura 3.13 – Aplicação de detecção por reflexão.	36
Figura 3.14 - Detecção por interrupção de feixe.	37
Figura 3.15 – Aplicação de detecção por interrupção de feixe.	37
Figura 3.16– Ilustração do sensor.....	38
Figura 3.17 – Descrição do circuito de acionamento dos LEDs	39
Figura 3.18– Descrição da porta de expansão	40
Figura 4.1 – Início da execução do programa	48
Figura 4.2 – Protoboard utilizado para montagem do sistema	51
Figura 4.3 – Protótipo inicial montado.....	52
Figura 4.4 – Simulação do semáforo de veículos.....	52
Figura 4.5 – Circuito do sensor	53
Figura 4.6 – Montagem dos semáforos	54
Figura 4.7 – Semáforos finalizados	55
Figura 4.8 – Sensor de presença finalizado	56
Figura 4.9 – Protótipo finalizado.....	57

LISTA DE QUADROS

Quadro 3.1 - Conector DB9 com designação de pinos.....	32
Quadro 4.1 – Custos do Projeto.....	57

LISTA DE ABREVIATURAS E SIGLAS

DB-9 – Conector Serial, padrão RS-232, com nove pinos

DENATRAN – Departamento Nacional de Trânsito

DETRAN-DF – Departamento de trânsito do Distrito Federal

DF – Distrito Federal

EPROM - Memória apagável somente de leitura programável

GND – Fio Terra

KIT – kit do PIC 16F628A

LED – Light Emitting Diodes - Diodo Emissor de Luz

OHM – Unidade de medida da resistência elétrica

RAM – Random Access Memory – Memória de Acesso Randômico

ROM – Memória apenas de leitura

RS-232 – Padrão da Interface Serial.

TTL – Transistor-Transistor Logic

Vcc – Fio de Alimentação

USART – Universal Synchronous Asynchronous Receiver Transmitter

CAPÍTULO 1: INTRODUÇÃO

Nas grandes cidades brasileiras, é crescente o número de interseções semaforizadas, tendo como objetivo proporcionar maior segurança à travessia de pedestres e veículos. Sabe-se, no entanto, que normalmente o aumento de segurança promove perda de fluidez na malha viária, resultando em longos tempos de viagem. Esse paradoxo vivido pelos engenheiros de tráfego é justificado em períodos de alta e média demanda de veículos, em que o número de conflitos aumenta bastante. Já em períodos de baixo volume de tráfego, uma grande quantidade de semáforos, quando não coordenados de forma eficiente, provoca tempos de espera desnecessários, gerados pelo número excessivo de paradas nas interseções, promovendo o desrespeito à sinalização semafórica e elevando o risco de colisões e atropelamentos. (PAIVA NETO; CASTRO NETO; LOUREIRO)

Quando o sistema de diminuição de tempo dos semáforos através do toque a um botão foi implementado em pontos estratégicos do DF, incluindo as comerciais do plano piloto, foco deste projeto, os pedestres ganharam uma forma prática e eficaz de diminuir o tempo de espera para atravessar com segurança. Porém, deficientes físicos cadeirantes e deficientes visuais não têm acesso a esse sistema, se mostrando necessária a criação de uma forma mais eficiente de fazer a mesma coisa, porém que seja acessível a toda população, incluindo crianças, deficientes físicos e visuais e, inclusive, animais de médio e grande portes, que podem causar acidentes sérios atravessando a rua com o sinal de trânsito de veículos aberto.

A proposta deste projeto é, basicamente, remover tal botão e incluir um sensor de presença, do tipo emissor-receptor, através de feixe de luz infravermelho, de modo que todos possam ter acesso ao mesmo. Tal idéia é ilustrada na figura 1.1.

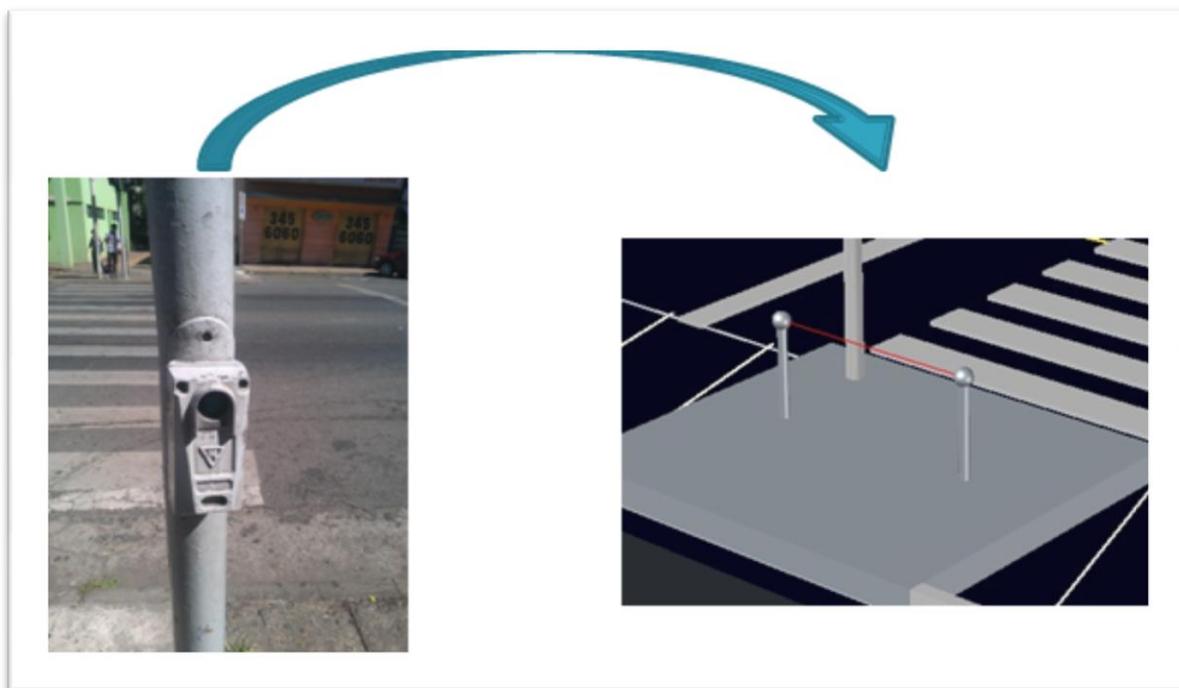


Figura 1.1 – Proposta do projeto

Fonte: Autor

1.1 MOTIVAÇÃO E POSICIONAMENTO

A melhoria de um sistema diariamente usado pela sociedade em seu direito de ir e vir contribui para a concretização de um projeto que tem em sua essência o objetivo de gerar facilidade de acesso para o público alvo, os pedestres.

A motivação para a realização deste projeto surgiu a partir da observação e, posteriormente, do estudo do grande número de pedestres que utilizam o sistema de diminuição de tempo, presente nos semáforos, todos os dias, em diversos pontos do Distrito Federal e em outras regiões do Brasil.

Com o intuito de aumentar a eficiência e a agilidade, tanto do trânsito de veículos como o de pedestres, este projeto propõe, por intermédio de um protótipo, a construção de um sistema de detecção de pedestres automatizado. Desta forma, foi proposto um sistema eficiente e inteligente que permite atravessar a rua com segurança, evitando assim perda de tempo desnecessária, tanto para os pedestres como para os motoristas.

1.2 OBJETIVOS GERAIS

O objetivo deste projeto é desenvolver uma solução de hardware e software que permita controlar o fluxo da faixa de pedestre e, conseqüentemente, o fluxo de veículos nas comerciais do plano piloto, com o uso de sensor de presença de pedestres em tais semáforos.

O projeto visa ajudar todos aqueles que possuem deficiência física ou visual, pois os mesmos não terão que fazer esforço algum para o fechamento do sinal de forma mais rápida e eficiente, além de beneficiar a todos que precisam atravessar a rua de forma segura e diminuir o tempo de espera dos pedestres com o fechamento do sinal mais rápido.

Neste projeto, para a detecção dos pedestres, foram utilizados sensores de presença que funcionam através de um transmissor fototransistor e um receptor infravermelho. Estas informações serão enviadas a um microcontrolador PIC.

O software desenvolvido para a detecção de pedestres indica, através de um bip, quando o pedestre parado na faixa é identificado pelo infravermelho. Com base na detecção referida, transmitirá os dados para o microcontrolador, que dará início ao processo. Também foi implementado ao projeto um sistema sonoro que auxilia os deficientes visuais a identificar a hora correta de atravessar a rua com segurança.

1.3 OBJETIVOS ESPECÍFICOS

Para o funcionamento do projeto, algumas tarefas precisaram ser executadas, tais como:

- Construir um protótipo simulando um sinal de trânsito;
- Criar um programa em Assembly para controlar todo o sistema de detecção e acionamento dos LEDs;
- Utilizar um PIC 16F628A para gerenciar a parte lógica do projeto;
- Utilizar sensores infravermelhos para detectar pedestres querendo atravessar a rua;

- Utilizar LEDs para a indicação do estado do sinal de trânsito, na cor vermelha indicando 'Fechado', na cor amarela 'Atenção' e na cor verde 'Aberto';
- Utilizar LEDs para a indicação do estado do sinal de pedestres, na cor vermelha indicando 'Fechado' e na cor verde 'Aberto';
- Elaborar um circuito que controle o sistema sonoro do projeto.

Já na parte de software, foi desenvolvido:

- Linguagem Assembly para programar o microcontrolador;
- Linguagem Assembly para programar o sensor de presença.

1.4 JUSTIFICATIVA E RELEVÂNCIA DO TEMA

Por priorizar o direito de ir e vir não apenas de pedestres que gozam de plena forma física, mas também deficientes físicos, visuais, idosos e crianças, este projeto vem de encontro com uma necessidade e uma obrigação, não apenas por parte do governo, mas pela população como um todo.

Na atualidade, a integração de pessoas com necessidades especiais vem se tomando um foco cada vez maior por parte do governo e da população como um todo, que se sensibiliza com a situação dessas pessoas que enfrentam problemas de acessibilidade em diversos locais, inclusive na simples ação de atravessar uma faixa de pedestres com semáforos com segurança.

A lei 10.098/2000 estabelece critérios básicos e normas gerais para a promoção da acessibilidade das pessoas portadoras de deficiência. Com relação aos semáforos, ela enuncia (grifo nosso):

Os semáforos para pedestres instalados nas vias públicas **deverão estar equipados com mecanismo que emita sinal sonoro suave, intermitente e sem estridência, ou com mecanismo alternativo**, que sirva de guia ou orientação para a travessia de pessoas portadoras de deficiência visual, se a intensidade do fluxo de veículos e a periculosidade da via assim determinarem.

No projeto em questão, foi implementado um mecanismo alternativo, de grande valia, barato, moderno e eficiente na locomoção de um deficiente visual, não apenas em entrequadras do plano piloto, mas também podendo ser implementado em sinais de grande fluxo de veículos e pedestres, ajudando não apenas deficientes visuais, mas também cadeirantes, que tem dificuldade de acesso as botoeiras presentes nos semáforos. Botões estes que tem o mesmo objetivo de diminuir o tempo de espera de pedestres, mas seu contato pode oferecer diversos riscos aos usuários por uma possível contaminação de bactérias, uma vez que os mesmos não são limpos.

O custo de implementação do projeto é baixo, o que viabiliza a integração e produção do mesmo, a nível industrial, podendo ser implementado em todos os semáforos presentes no plano piloto. O uso de microcontroladores da Microchip Technology, em especial o PIC 16F628A, contribuiu enormemente para o baixo custo do mesmo.

1.5 DIAGRAMA DE BLOCOS

O diagrama de blocos tem o objetivo de representar de forma simples, porém objetiva, a composição geral do projeto.

A figura 1.2 ilustra o diagrama de blocos deste projeto onde se pode observar o fluxo do projeto.

O sistema executa a condição principal, cujo sinal de trânsito permanece aberto por um tempo mínimo de 20 segundos e, após decorrido tal tempo, o sistema verifica a presença de pedestres na calçada, através do sensor infravermelho.

Logo, quando o receptor não recebe o sinal do sensor significa que não há pedestres querendo atravessar a rua. Caso esta condição não se altere, o sistema fica em estado de espera com o sinal de trânsito liberado. Após o sensor identificar um pedestre, o sistema ativa sua rotina de funcionamento, mudando o estado do sinal de trânsito para atenção e em seguida para fechado e liberando a passagem de pedestres, ativando a luz verde.

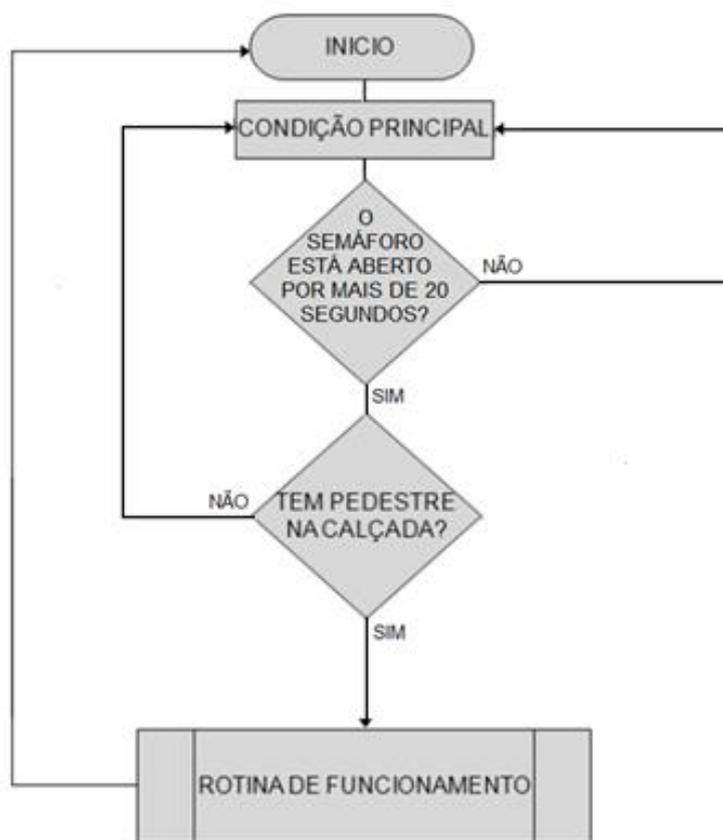


Figura 1.2 – Fluxograma do software

Fonte: Autor

1.6 ESCOPO DO TRABALHO

O protótipo utiliza sensor de presença dos dois lados da rua que tem como base um emissor de luz infravermelha com um receptor em linha, que detecta a presença de pedestres no semáforo através da interrupção do feixe de luz infravermelha, acionando o sistema, tendo como base as regras especificadas no capítulo seguinte desta monografia.

Não foi implementado no protótipo a função de detectar quantos carros há na pista, apenas se há ou não pedestres querendo atravessá-la. O objetivo focal do projeto é identificar que há pedestres querendo atravessar a pista para diminuir o tempo de espera dos mesmos e trazer mais segurança a todos, uma vez que o sistema obriga o pedestre a ir até a faixa, já que o sinal de trânsito não fechará se não houver pessoas querendo atravessar a rua.

Na figura 1.3 temos a estrutura básica do trabalho.

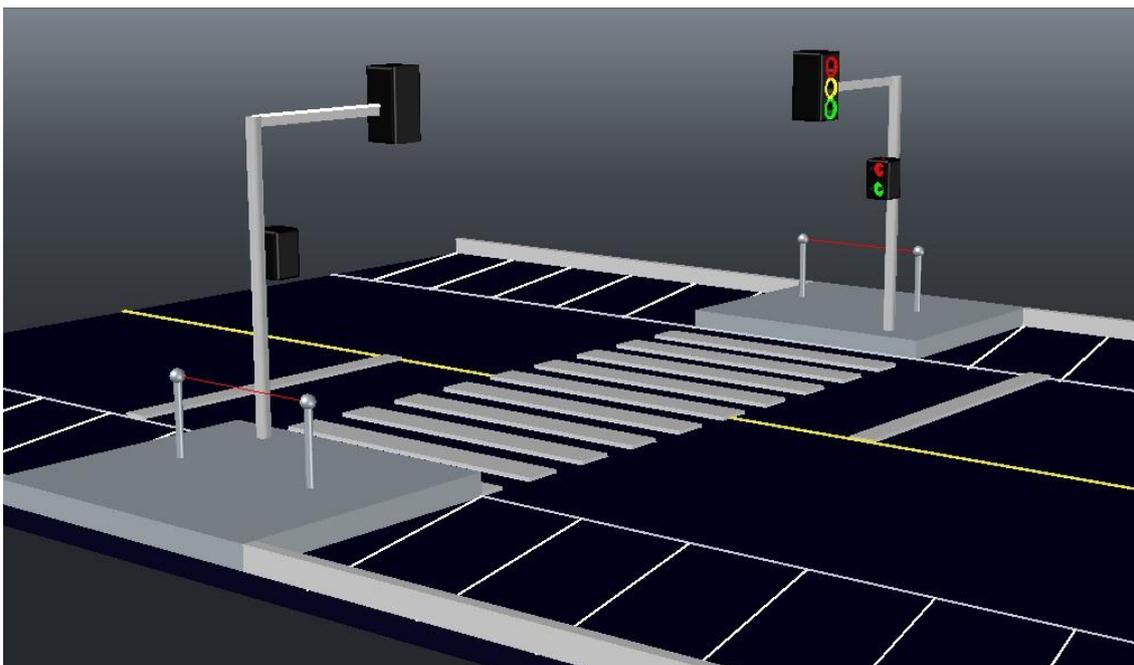


Figura 1.3 – Estrutura do trabalho

Fonte: Autor

A figura 1.3 foi desenhada no Autocad 2012 e retrata a estrutura básica dos semáforos do plano piloto com o sensor de presença desenvolvido nesta monografia. Pode-se perceber que os postes com sensores ficaram instalados em pontos estratégicos da travessia de pedestres, de modo que apenas pessoas que realmente tem a intenção de atravessar a rua interceptarão o feixe de luz infravermelho.

Para definição de tais parâmetros, foi feita uma pesquisa em quadras comerciais do plano piloto, sendo identificado que pessoas que tem o intuito de atravessar a rua ficam posicionadas exatamente na posição aonde os sensores foram instalados. Foi observado também, que motoristas que estacionam seus veículos ao lado do local de travessia de pedestres, costumam subir na calçada, mas nem sempre atravessam a rua. Para evitar uma possível detecção desnecessária do infravermelho, os postes com os mesmos foram instalados com certa distância, dando a possibilidade para o motorista estacionar e descer de seu carro sem acionar o sistema.

Na figura 1.4 a seguir, pode-se observar o tamanho dos postes com infravermelho, a distância do poste às vagas próximas e a distância entre o emissor e o receptor.

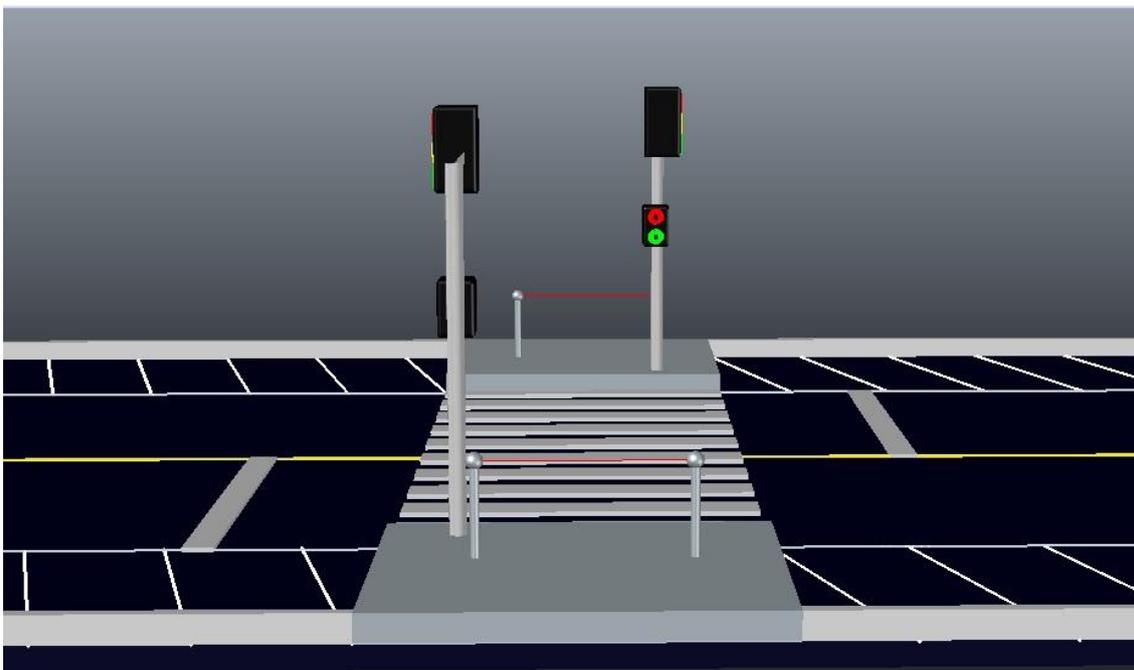


Figura 1.4 – Distâncias do sensor

Fonte: Autor

No projeto em questão, a altura dos postes contendo os sensores emissor-receptor ficou determinada em aproximadamente oitenta centímetros, uma vez que animais de médio porte como cachorros tem uma altura aproximada de 85cm. Foi observado também que 20% das pessoas que atravessam a rua em comerciais do plano piloto são crianças. Por fim, foi estudado e observado que cadeirantes ficam, em média, com altura de 1,10 a 1,40 do solo.

Já a distância entre as vagas de veículo e os postes com os emissores ficou determinada em aproximadamente 1,30 metros. Tal distância foi determinada após a verificação que a maioria dos motoristas que estacionam em tais vagas costumam descer de seus veículos e imediatamente subir na calçada próxima ao sinal de trânsito. Com isso, tais pedestres poderiam interromper o feixe de luz infravermelha se os mesmos fossem instalados nas extremidades da calçada, além de não permitir a abertura total da porta do veículo.

Por fim, a distância dos postes contendo os emissores à pista, ficou determinada em aproximadamente um metro, distância segura para não ocorrer qualquer tipo de acidente entre pedestres e veículos.

1.7 PROBLEMA A SER RESOLVIDO

O problema estudado neste projeto de pesquisa tem como objetivo apresentar uma solução viável e barata para evitar os transtornos causados pela perda de tempo desnecessária nos semáforos, tanto para os motoristas quanto para os pedestres.

Esse projeto visa beneficiar pedestres que gozam de plena saúde física, cadeirantes, crianças e deficientes visuais que não conseguem ter acesso ao botão de acionamento de redução do tempo de espera em semáforos do plano piloto, além de evitar uma possível transmissão de doença às pessoas que usam o mesmo, por uma possível contaminação de bactérias, uma vez que os mesmos não são limpos e são fabricados em plástico, que retém sujeira e organismos passíveis de transmissão de doenças através do simples contato do ser humano com o mesmo.

1.8 METODOLOGIAS

Para a realização da montagem do protótipo, foram utilizadas diversas fontes bibliográficas sendo as principais: livros, revistas, sites de internet, monografias e artigos científicos. O trabalho também reúne conteúdos ministrados nas disciplinas do curso de graduação da Engenharia de Computação tais como:

Circuitos e Máquinas Elétricas, Sistemas de comunicação, Linguagem e Técnica de Programação, Circuitos eletrônicos, Microprocessador e Microcontroladores, Instalações Elétricas, Arquitetura de Computadores.

1.9 ESTRUTURA DA MONOGRAFIA

Esta monografia é composta de 5 capítulos, iniciando com a INTRODUÇÃO, que apresenta a motivação do projeto, suas restrições, os principais objetivos, a metodologia de pesquisa, o diagrama de blocos e também toda estrutura desse trabalho.

No capítulo 2 são apresentados os FUNDAMENTOS TEÓRICOS, com descrição dos principais assuntos abordados e o funcionamento detalhado de sinais de trânsito.

No capítulo 3 encontra-se a DESCRIÇÃO DO HARDWARE utilizado em todo o projeto.

No capítulo 4 é apresentado o detalhamento dos TESTES E IMPLEMENTAÇÃO da proposta de resolução.

O capítulo 5 apresenta a CONCLUSÃO. Este capítulo marca o encerramento da monografia, com a avaliação global do projeto e as dificuldades encontradas.

CAPÍTULO 2: FUNDAMENTOS TEÓRICOS

2.1 CONTROLES DE TRÂNSITO

A autorização e a proibição de movimentos para os veículos de uma corrente de tráfego são feitas por equipamentos instalados nos cruzamentos e estes são denominados de semáforos. (DENATRAN, 1984)

O semáforo é um instrumento de controle modificável que informa as prioridades de passagem para motoristas e pedestres em uma via ou em uma interseção de vias. Para indicar esta autorização ou proibição do movimento de passagem, utilizam-se focos luminosos afixados em grupos ao lado da via ou suspensos sobre ela, através de elementos de sustentação (postes). De uma forma geral, durante a implantação de um *semáforo*, faz-se necessário a implantação das sinalizações horizontais (faixa de pedestre) e verticais. (BARBACENA, 1994)

O termo semáforo de tráfego é considerado uma instalação completa, incluindo os sinais luminosos (porta-foco, lentes, lâmpadas, coluna, braço projetado, etc.), os fios elétricos, o *controlador de tráfego*, etc. (*idem*)

Existem dois tipos de semáforos; os veiculares e os de pedestres, mas neste projeto o foco será o semáforo veicular.

2.2 SEMÁFORO VEICULAR

Segundo DENATRAN, o semáforo veicular é um dispositivo geralmente composto de três focos de luz de cores distintas, um de cor vermelha, outro de cor amarela e, por último, um de cor verde. Essas cores são padronizadas internacionalmente e têm funções específicas, regulamentadas no Código de Trânsito Brasileiro, conforme descrito a seguir:

- *Verde*: os condutores de veículos podem seguir em frente, virar à esquerda ou direita, a menos que estejam impedidos fisicamente por outro dispositivo de controle de tráfego ou autoridade legal;

- *Amarelo*: os condutores devem parar o veículo antes de entrar na região de cruzamento e permanecer parados até receber autorização de seguir em frente através da luz verde ou por uma autoridade legal. Caso não seja possível realizar a parada do veículo em segurança ou se não houver tempo hábil para realizar a parada, o condutor deve seguir em frente e cruzar a interseção;
- *Vermelho*: os condutores devem parar os veículos antes de entrar na interseção e permanecer parados até receber autorização para seguir em frente, seja pela sinalização da luz verde ou por uma autoridade legal.
- *Amarelo Piscante*: Este estado é utilizado normalmente simultaneamente em todas as fases semaforicas, sendo ativada quando o controlador está com problemas ou está em manutenção. É muito comum também usar este estado em horários de baixo volume de tráfego, normalmente após a meia noite.

O principal objetivo do semáforo veicular é permitir ou proibir a passagem do tráfego através das indicações luminosas verde/vermelho, respectivamente.

Porém, para não proceder a uma interrupção brusca de movimento, criou-se o tempo de atenção, que é a situação intermediária entre movimento e parada, representado pela cor amarela. Ao receber a indicação amarelo, os motoristas são alertados sobre a proximidade da mudança, porém tem tempo suficiente para reagir a ela.

Os tempos de cada indicação luminosa de um semáforo são denominados estágios ou intervalos, enquanto a seqüência de luzes (verde, amarelo e vermelho) em cada aproximação é denominada fase. O tempo total para a completa seqüência luminosa em todas as aproximações é denominado ciclo. E o tempo entre o fim do verde de uma fase (perda do direito de passagem) e o início de outra é denominado período entreverdes e geralmente possui a mesma duração do estágio amarelo.

[DENATRAN, 1984]

2.3 CÓDIGO DE TRÂNSITO BRASILEIRO

O atual código de trânsito brasileiro foi promulgado em 23 de Setembro de 1997. Na época de sua promulgação, ele causou muita polêmica por causa da grande severidade com relação às infrações principalmente no que dizia respeito ao valor das multas.

Os pedestres foram bastante beneficiados com esse novo código porque passaram a ser mais respeitados no trânsito, em contrapartida, tiveram mais direitos e deveres para cumprir podendo até serem multados se não cumprissem certas regras.

A lei de respeito à faixa de pedestres foi promulgada em abril, mas o novo código também contribuiu bastante para que essa lei realmente fosse respeitada.

No novo código, as penalidades para qualquer motorista que desrespeitar a faixa de travessia são extremamente severas e consideradas graves ou gravíssimas, tirando cinco ou sete pontos da carteira do motorista infrator e com uma multa com valor elevado. Entre essas infrações podemos destacar: estacionar sobre a faixa de travessia, não dar preferência ao pedestre, fazer ultrapassagem perto da faixa de travessia.

Os pedestres também devem seguir algumas regras no trânsito para sua segurança. Entre essas regras podemos destacar: o pedestre deve sempre andar em local apropriado, o pedestre é obrigado a atravessar em local apropriado, se este estiver a menos de cinquenta metros de distância de onde ele se encontra. Se não houver passagem preferencial ele deve aguardar em local apropriado o momento correto de atravessar a via com segurança.

Ao atravessar uma via, o pedestre deve manter uma velocidade constante e atravessar perpendicularmente a via, independentemente se está atravessando na faixa ou não, fazendo assim o percurso mais curto.

Em faixas de travessia sem semáforo, o pedestre deve pedir passagem fazendo o “sinal de vida”, aguardar que os carros parem e depois efetuar a travessia, os veículos só podem entrar em movimento depois que o pedestre chegar ao outro lado da via. Nos semáforos, o pedestre deve aguardar o sinal verde para atravessar, e mesmo que o sinal fique verde para os veículos, estes devem esperar o fim da travessia dos pedestres para entrar em movimento. (DETRAN-DF)

2.4 TERMOS DA TEMPORIZAÇÃO

Para entendermos melhor os termos utilizados na temporização dos semáforos, segue abaixo uma descrição de cada item. (BARBACENA, 1994)

2.4.1 Fase

É uma sequência completa de indicação de cores dos focos, que permite a um conjunto de vias o controle simultâneo do direito de passagem. No caso da fase veicular temos fisicamente três focos, e uma fase é completada após passar pela sequência verde, amarelo e vermelho. No caso da fase de pedestre temos dois focos, e uma fase é completada após passar pela sequência verde, vermelho piscante e vermelho.

2.4.2 Ciclo ou tempo de ciclo

É o tempo gasto para completar uma fase semafórica. A partir deste período, começa a repetir periodicamente, as cores dos focos desta fase.

2.4.3 Estágio

É formado por intervalos dentro de um ciclo, onde são alternados os movimentos das correntes de tráfego. Isto significa que cada estágio corresponde do início ao fim do verde, para cada fase semafórica, em um mesmo cruzamento.

2.4.4 Período de Entreverdes

É período entre o término de verde de uma fase e o início do tempo de verde da outra fase. Este período é caracterizado por ser curto e normalmente, no Brasil, é igual ao tempo de amarelo e em alguns casos, compõe-se do tempo de amarelo mais o tempo de vermelho total.

2.4.5 Vermelho Total

É um período durante o qual todos os focos das fases semafóricas ficam com a cor vermelha. Este período é utilizado para garantir uma maior segurança na limpeza dos veículos no cruzamento e/ou criar um tempo especial para travessia de pedestres.

CAPÍTULO 3: DESCRIÇÃO DO HARDWARE

3.1 MICROCONTROLADOR PIC 16F628A

O PIC16F628a é um microcontrolador fabricado pela Microchip Technology com as seguintes características:

- composto de 18 pinos;
- possui somente 35 instruções no seu microcódigo;
- sinal de clock de frequência até 20 MHz;
- memória de programa do tipo Flash de 2048 words (1 word = 32 bits);
- 224 bytes de memória RAM para dados;
- 128 bytes de memória EEPROM para dados;
- instruções de 14 bits com 200ns de tempo de execução;
- dados de 8 bits por endereço de memória;
- 15 registradores especiais;
- 16 pinos os quais podem ser configurados como entrada e/ou saída.

A pinagem do microcontrolador PIC 16F628A é apresentada na figura 3.1

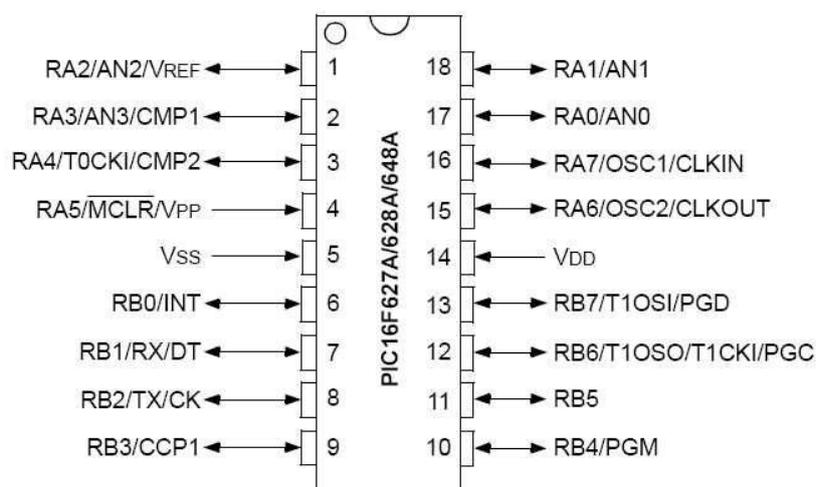


Figura 3.1 - Pinagem do PIC 16F628A

Fonte: [www.microchip.com]

Os pinos de RA0 a RA7 e de RB0 a RB7 podem ser configurados como entradas ou saídas digitais. A alimentação se dá no pino Vdd, ligado normalmente em 5V com faixa de tolerância de 2 a 6V e o pino Vss é a referência de terra. O pino OSC1/CLKIN é utilizado para sinal de clock produzido por cristal ou um circuito externo e o pino OSC2/CLKOUT para sinal de clock por cristal (utilizado em conjunto com OSC1/CLKIN). O pino MCLR é uma entrada de sinal de reset em nível baixo (zero). O PIC16F628a é composto pelos subsistemas digitais conforme diagrama de blocos, apresentado na figura 3.2 abaixo:

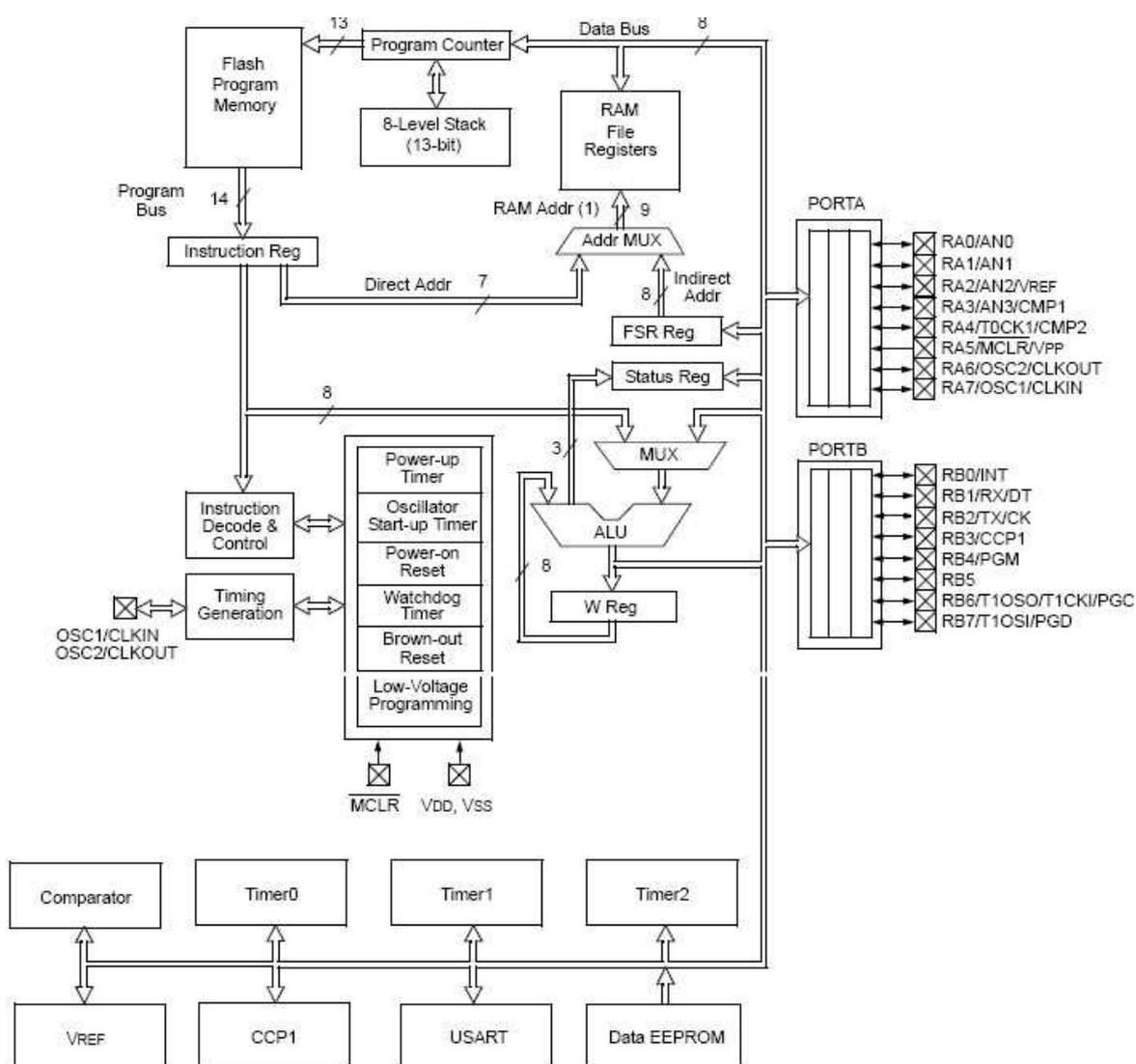


Figura 3.2- Diagrama de blocos do PIC 16F628A

O PIC16F628a possui uma arquitetura denominada Harvard para acesso aos dados e às instruções do programa, a qual é caracterizada por utilizar um bus (via de acesso) para comunicação dos dados e outro bus distinto para comunicação com o programa. A maioria dos microcomputadores e microcontroladores no mercado utilizam um único bus, tanto para dados quanto para instruções do programa. Nota-se a diferença no uso desta arquitetura na visualização da memória do microcontrolador PIC: existe uma memória para o programa e outra para os dados a serem utilizados.

3.2 ORGANIZAÇÃO DA MEMÓRIA

Os microcontroladores da linha PIC possuem pinos que podem ser configurados como entrada ou saída digital. Cada conjunto de oito destes pinos é denominado Port, geralmente classificado por uma letra do alfabeto (PortA, PortB, PortC,...). Para isto, existem implementados na memória RAM do PIC registradores especiais para configurar e ler/modificar o valor destes pinos. No PIC16F628a, dois destes registradores, o TRISA e o TRISB, são responsáveis pela configuração destes pinos para funcionamento como entrada ou saída digital. A abreviatura TRIS para esses registradores provem de tri-state (três estados) porque alguns pinos podem funcionar de três modos diferentes: como entrada digital, saída digital ou com outra função específica. A figura 3.3 ilustra a estruturação do PIC em questão.

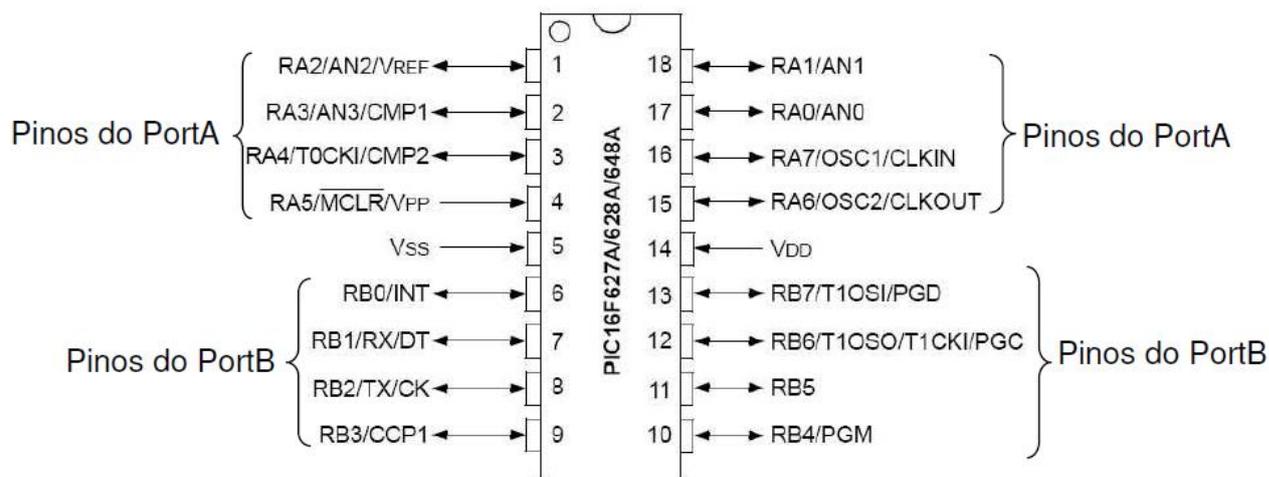


Figura 3.3- Estruturação do PIC 16F628A

Fonte: [www.microchip.com]

3.3 CIRCUITO SONORO

Para a sonorização do sistema, que tem o objetivo de avisar/alertar os deficientes visuais, além dos pedestres desatentos, quando o sinal de trânsito estiver aberto para sua passagem em segurança. Foi utilizado um sistema sonoro chamado *buzzer*. Este consiste basicamente em um pequeno *hardware*, do tamanho de uma moeda, composta de duas camadas de metal e uma camada interna de cristal piezoelétrico (como um sanduíche). Ao ser alimentado com uma fonte de sinal, através do microcontrolador, vibra na mesma frequência recebida, funcionando como uma sirene. Existem várias versões e tamanhos. Todos os dispositivos sonoros de alarmes (como os de automóvel) usam um *buzzer* para o som.

Sua vantagem em relação a altos-falantes comuns é que consome pouca energia, fundamental neste protótipo que preza pelo baixo custo com eficiência, em relação à potência sonora, sendo facilmente alimentado com a voltagem emitida pelo microcontrolador quando o mesmo envia nível lógico um ao *buzzer*.



Figura 3.4 – Circuito de sonorização (*buzzer*)

Fonte: [www. icradenet.com]

Nesse projeto, a implementação de tal hardware foi simples, uma vez que o mesmo é alimentado pelo próprio kit. Para fazer tal implantação, foi necessário apenas indicar no código que tal dispositivo é de saída de dados, uma vez que o mesmo só entrará em funcionamento após iniciado o terceiro e último ciclo do processo (quando o sinal de veículos é bloqueado e o sinal de pedestres é liberado).

3.4 KIT DO MICROCONTROLADOR 16F628A

Para o desenvolvimento deste projeto, foi adquirido o kit de desenvolvimento ACEPIC 18, da ACEPIC, por apresentar todos os requisitos necessários para o projeto. O kit possui as seguintes características:

- Display de Cristal Líquido (LCD) 16 colunas e 2 linhas com Backlight;
- Display de 7 segmentos;
- Led`s
- Trimpots para comparadores;
- USART (RS232);
- Gravação OnBoard;
- Botões de entrada ou interrupção externa e Timer0 e Timer1 (simulação de sinal externo);
- Conexão para gravação in-circuit (ICSP);
- Saída para todas as portas do microcontrolador.

O microcontrolador utilizado nesse kit é o PIC 16F628A, que é totalmente compatível com o objetivo do projeto em questão. Ele é composto por uma fonte de alimentação de 12V – 400mA e sua comunicação com o computador é realizada através de um cabo serial no padrão RS-232, utilizado para comunicação e gravação. O esquemático dele encontra-se na figura 3.5.

Tal kit foi escolhido por ser de simples implementação, por ser barato e por ser popular, facilitando a consulta a materiais específicos.

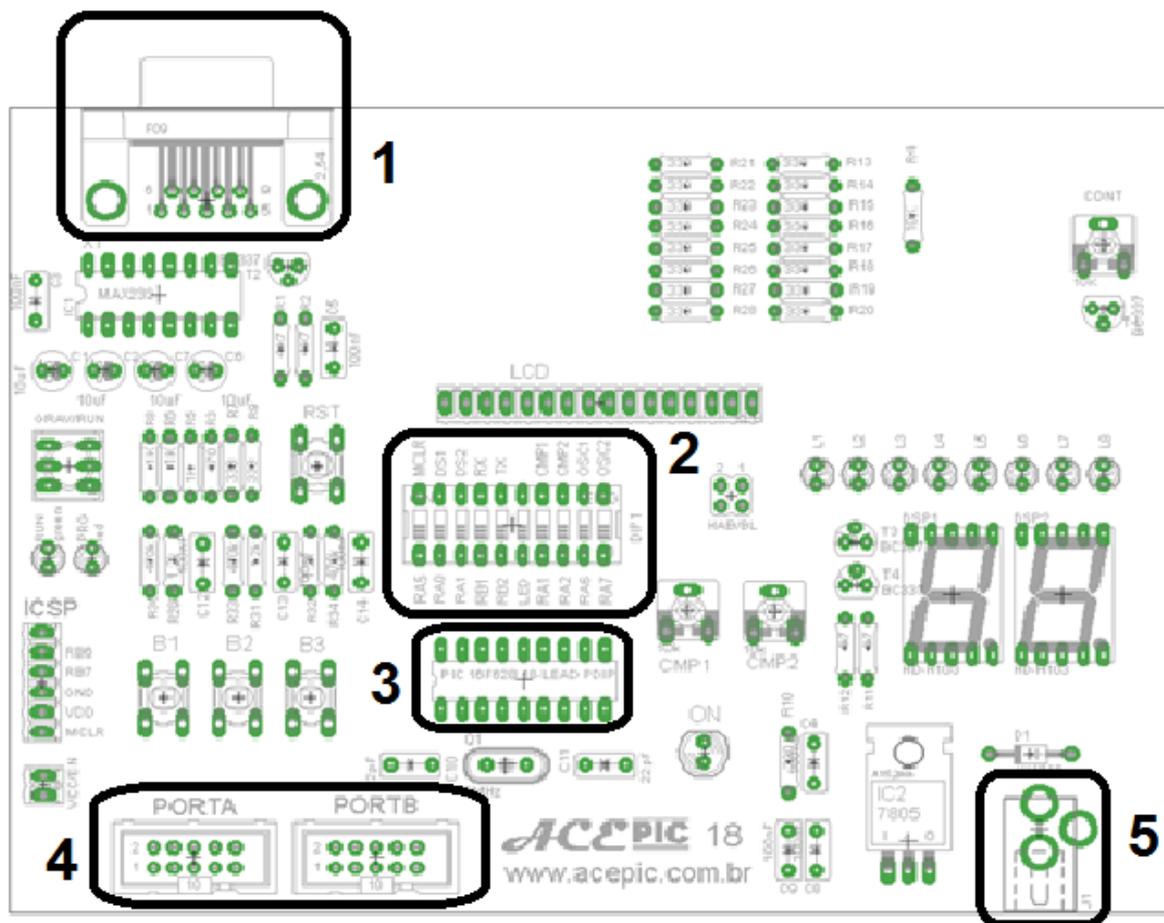


Figura 3.5 – Layout da placa ACEPIC 18

Fonte: [Manual ACEPIC 18]

Aonde:

Número 1 → Entrada do cabo RS 232 ao kit, necessário para a transmissão do código ao microcontrolador.

Número 2 → Interruptor do KIT, aonde é configurado os dispositivos do kit que devem permanecer ligados ou desligados.

Número 3 → Microprocessador 16F628A, que gerenciará a parte lógica do projeto.

Número 4 → Portas de entrada e saída do PIC, aonde serão conectados todos os LEDs necessários para o funcionamento dos semáforos, além do sensor de presença e do *buzzer*.

Número 5 → Entrada de energia necessária para alimentar o KIT.

Para o correto funcionamento desse kit de desenvolvimento ele deve ser utilizado e alimentado com uma fonte de 5V. A fonte utilizada é de 12V por 400mA conectado ao kit. Para utilização desta fonte com o microcontrolador foi necessário a utilização de um regulador para baixar a tensão de 12 V para 5 V. O próprio kit de desenvolvimento realiza internamente a regulação desses 12V para os 5V necessários. A figura 3.6 ilustra a imagem dessa fonte usada no projeto.



Figura 3.6 – Fonte do Kit

Fonte: Autor

3.5 REGULADOR DE TENSÃO 7805

O regulador de tensão utilizado foi o de número 7805 da série 78XX. Ele regula a tensão do kit de desenvolvimento, de 12 V para 5V, alimentando todo o sistema. Esse regulador contém 3 pinos: o de entrada, o de saída e o aterramento. A figura 3.7 ilustra o regulador 7805 e seus pinos.

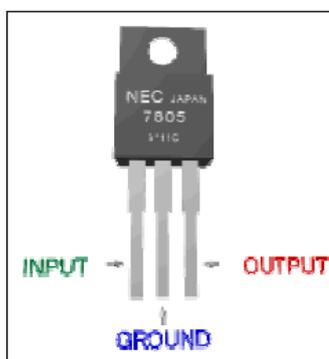


Figura 3.7 – Descrição do circuito de acionamento dos LEDs

Fonte: [Manual ACEPIC 18]

Pino 1: INPUT - Entrada que recebe a tensão de 12V;

Pino 2: GROUND – Terra;

Pino 3: OUTPUT - Saída regulada de 5V.

3.6 COMUNICAÇÃO SERIAL

A comunicação serial é o processo de enviar bits de cada vez, sequencialmente, num canal de comunicação ou barramento. O dispositivo periférico que acompanha o kit de desenvolvimento é a porta serial DB9, ou seja, o padrão RS232, como mostra a figura 3.8.

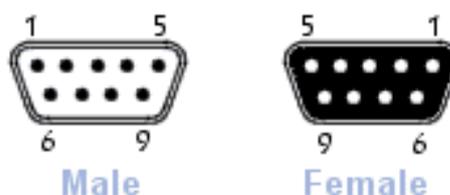


Figura 3.8 – Conector DB9 com sua pinagem enumerada.

Fonte: KIOSKEA, 2011

3.7 PINAGEM DO CONECTOR SERIAL

Segundo o padrão RS232, cada pino de um conector serial tem uma função específica de acordo com o Quadro 3.1, que deve ser seguido para a correta montagem de um cabo de comunicação.

Número	Nome	Designação
1	CD – Carrier Detect	Detecção de portador
2	RXD – Receive Data	Recepção de dados
3	TXD – Transmit Data	Transmissão de dados
4	DTR – Data Terminal Ready	Terminal pronto
5	GNS – Signal Ground	Terra
6	DSR – Data Set Ready	Dados prontos
7	DSR – Request to Send	Pedidos de emissão
8	CTS – Clear to Send	Limpar para enviar
9	RI – Ring Indicator	Indicador de campainha

Quadro 3.1 - Conector DB9 com designação de pinos.

Fonte: SYSTEM, 2007

Neste projeto foram utilizados apenas três pinos. O pino 2, RXD (Receive Data), o pino 3 TXD (Transmit Data) e o pino 5 GND (Sinal Ground)

3.8 CONVERSOR DE NÍVEL SERIAL PARA TTL

Para comunicação entre o computador e o microcontrolador, através da porta serial, foi utilizado o circuito integrado MAX232, ilustrado na figura 3.10, que é um conversor de sinais RS232 para TTL. Sendo o protocolo de comunicação RS-232 usado pelo CI (Circuito Integrado) o processo de enviar dados, um bit de cada vez, sequencialmente em um canal de comunicação ou barramento. A figura 3.9 é uma foto do MAX232 utilizado no projeto.

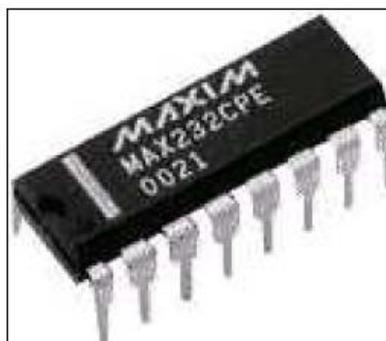


Figura 3.9- MAX232.

Fonte: CIRCUIT SHOP, 2012

A figura 3.10 ilustra a pinagem do MAX232.

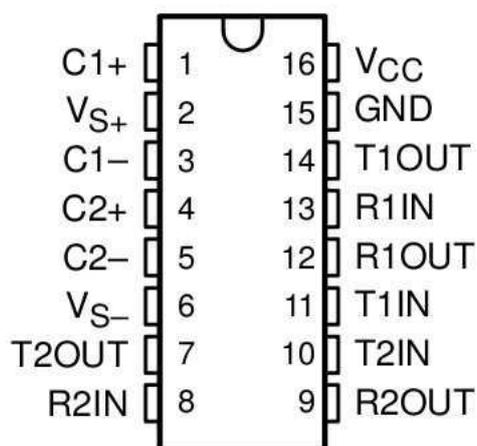


Figura 3.10 – Pinagem do MAX232.

Fonte: TEXAS INSTRUMENTS, 2004

Pino 1 e 3: capacitor de $1\mu\text{F}$;

Pino 2 e 16: capacitor de $1\mu\text{F}$;

Pino 4 e 5: capacitor de $1\mu\text{F}$;

Pino 6: capacitor de $1\mu\text{F}$ aterrado;

Pino 7: não utilizado;

Pino 8: não utilizado;

Pino 9: não utilizado;

Pino 10: não utilizado;

Pino 11: Ligação com PIC 16F876A no pino 17 (TX).

Pino 12: não utilizado;

Pino 13: Ligação com a porta serial no pino 3 (TXD).

Pino 14: não utilizado;

Pino 15: terra;

Pino 16: fonte de alimentação de 5V e Capacitor de $10\mu\text{F}$ para desclopamento.

A figura 3.11 a seguir ilustra a conexão da porta serial RS 232, usada para a inclusão de dados no PIC, com o microcontrolador e a função do MAX232 nesta conexão.

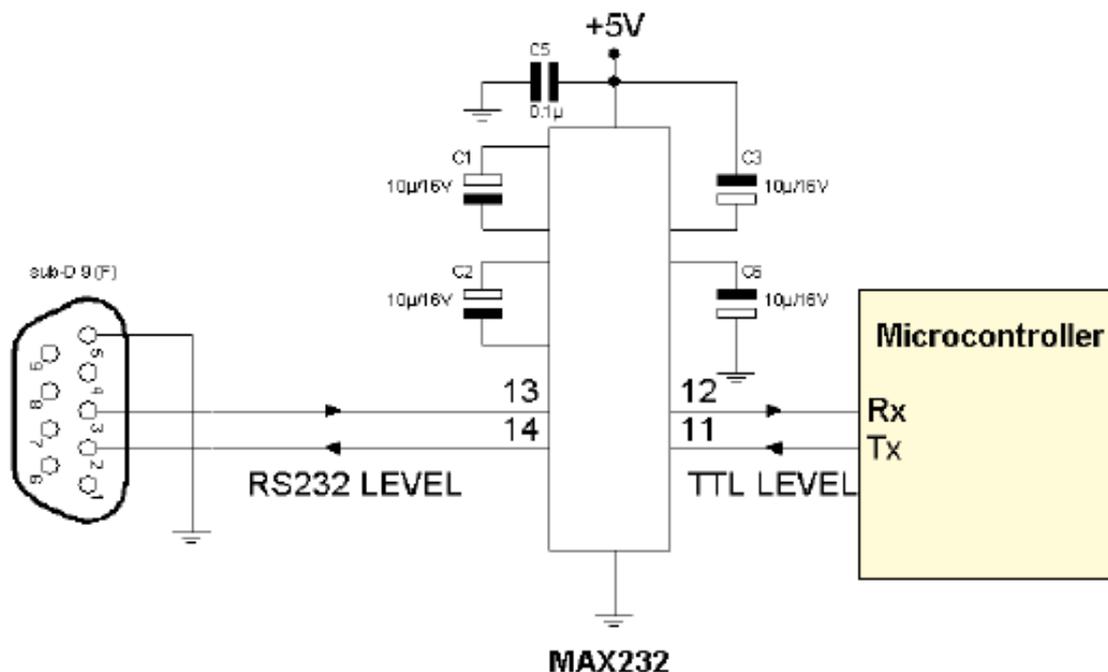


Figura 3.11 – Porta serial – MAX232 – Microcontrolador

Fonte: [Manual ACEPIC 18]

3.9 INFRAVERMELHO

Por ser uma tecnologia de simples implementação, o sinal infravermelho foi escolhido para detectar a presença de pedestres nos semáforos das comerciais do plano piloto no protótipo montado. Essa tecnologia possui a vantagem de ser de baixo custo e de não necessitar de contato físico. Neste projeto, foi usado a detecção por interrupção de feixe, explicado em detalhes abaixo.

3.10 TIPOS DE DETECÇÃO E APLICAÇÕES

Os tipos de detecção podem ser divididos em dois grupos: detecção por reflexão e detecção por interrupção de feixe.

- Detecção por reflexão: Nesse tipo de detecção, a luz emitida pelo emissor cria uma região ativa cuja presença de um objeto faz com que a luz seja refletida de volta para o receptor. Se o sinal alcançar um limite pré-definido ativa o sensor indicando a presença do objeto. A figura 3.12 ilustra um exemplo de detecção por reflexão.

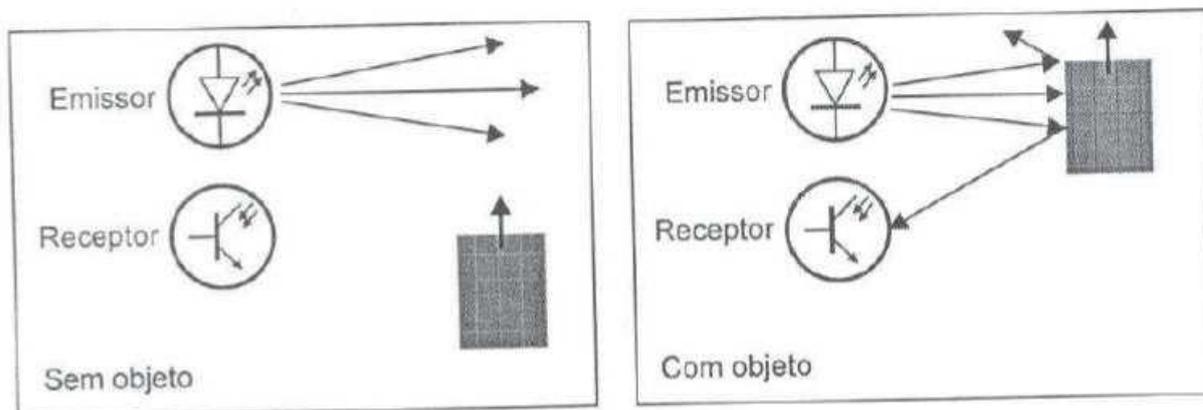


Figura 3.12 - Detecção por reflexão.

Fonte: THOMAZINI, 2005; URBANO BRAGA, 2005

A figura 3.13 ilustra o monitoramento de uma sala usando a detecção por reflexão. O detector monitora o padrão de reflexão na sala. Se alguém perturbar o padrão de reflexão, o detector de movimento envia um sinal de alarme para a caixa de controle.

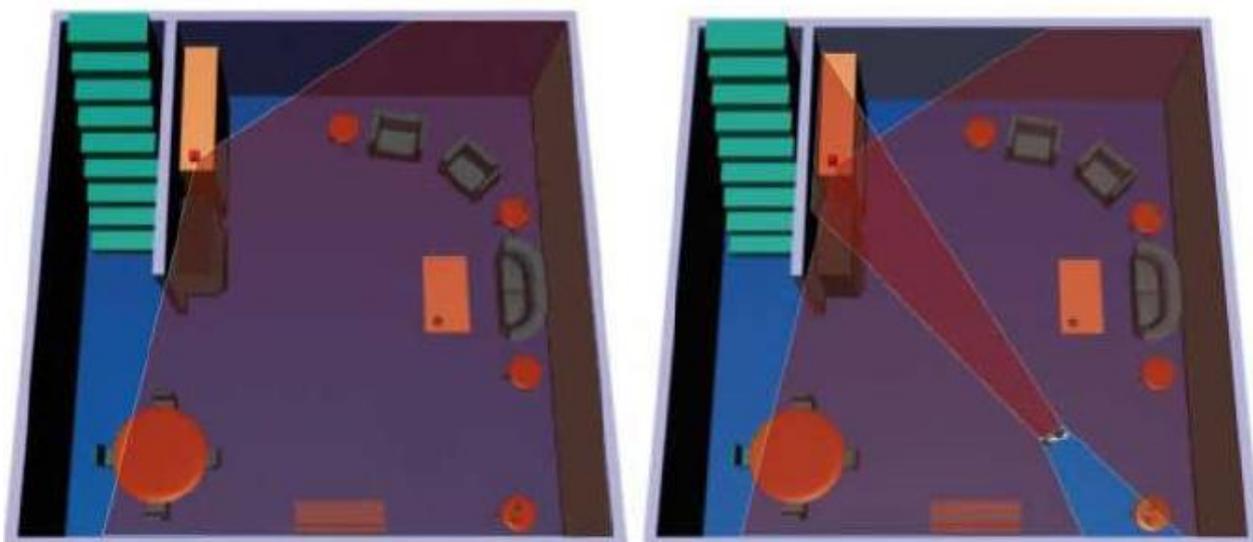


Figura 3.13 – Aplicação de detecção por reflexão.

Fonte: Radcomalarmes,2011

Detecção por interrupção de feixe: Neste tipo de aplicação, ao serem alinhados, o emissor e o receptor criam entre si uma barreira de luz. O receptor, então, fica constantemente recebendo o feixe de infravermelho. A presença de um objeto interrompendo essa barreira faz com que o sensor seja ativado. A Figura 3.14 ilustra este tipo de aplicação.

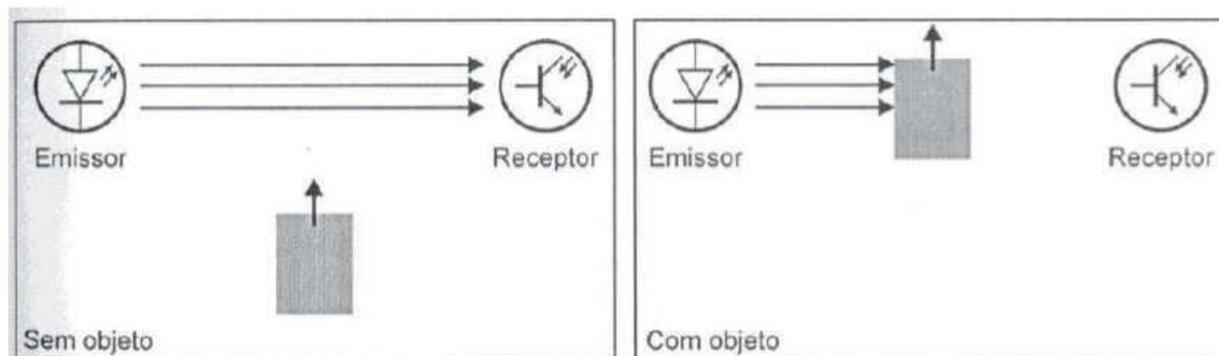


Figura 3.14 - Detecção por interrupção de feixe.

Fonte: THOMAZINI, 2005; URBANO BRAGA, 2005

Uma aplicação para esse tipo de técnica é a implementação de sistemas de segurança ou contadores de peças em uma esteira na linha de produção industrial, além de detectar entrada e saída de pessoas e automóveis. A Figura 3.15 ilustra a aplicação de detecção de entrada e saída de pessoas.



Figura 3.15 – Aplicação de detecção por interrupção de feixe.

Fonte: REDETEC, 2011

3.11 CIRCUITO INFRAVERMELHO

O sensor óptico utilizado se baseia em um simples transmissor e receptor de luz infravermelha, por meio de um fototransistor. O sensor escolhido possui um alcance testado de cerca de 1 metro, mais que o suficiente para a representação, e possui um baixo custo, uma vez que além do princípio óptico, se baseia no funcionamento de codificador e decodificador do próprio kit.

Neste sensor, ao ter seu emissor alinhado com o receptor, envia à saída aproximadamente 3V, sendo interpretado pelo kit como nível lógico 1 (um). Quando o feixe de luz entre emissor e receptor é interrompido por algum anteparo, o nível lógico é alterado para 0 (zero), pois a tensão enviada para o kit é zerada. Esses níveis lógicos (bits) são usados como informações para o microcontrolador processar a entrada dos dados com as informações de sua memória e colocar uma saída nos devidos pinos programados como tal.

Para detecção de pedestres no semáforo, tal circuito foi implantado na calçada que antecede o sinal de trânsito. A figura 3.16 ilustra o circuito emissor receptor.

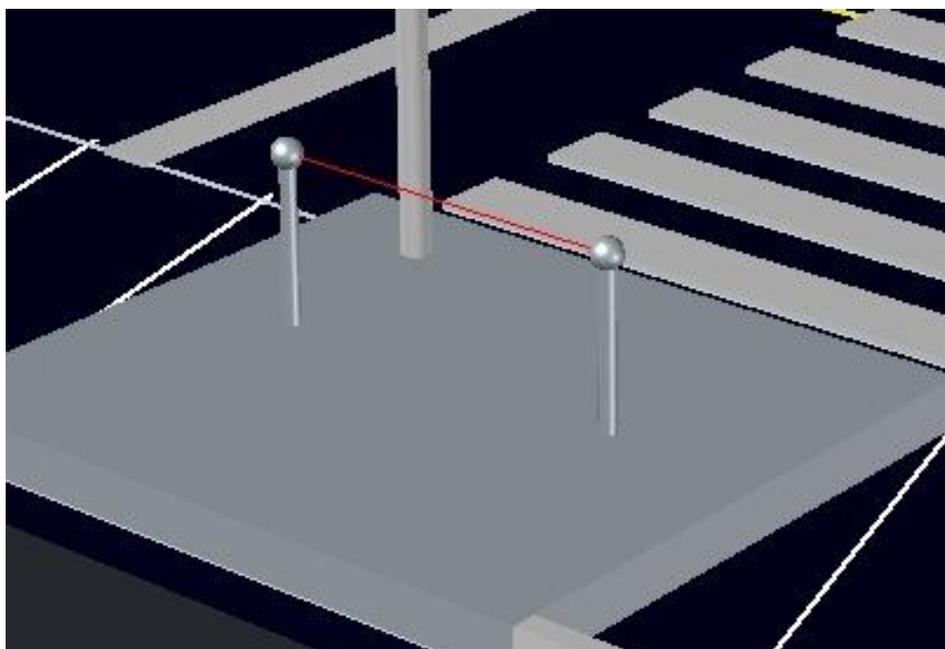


Figura 3.16– Ilustração do sensor

Fonte: Autor

A linha vermelha ilustra o sinal infravermelho que deverá ser interrompido para detecção do pedestre na calçada. Já as hastes ilustram a sustentação de tais sensores, que deverão estar a aproximadamente um metro do solo. Tais postes deverão ser fixos, perfeitamente alinhados, para o correto funcionamento do sistema em questão.

3.12 LED

Os cinco LEDs que compõem o kit, estão ligados na portA e a portB do microcontrolador e foram utilizados para simular as luzes do sinal de trânsito de do sinal de pedestres. O circuito presente no kit para acionar os LEDs está representado na figura 3.17, presente também no manual do kit:

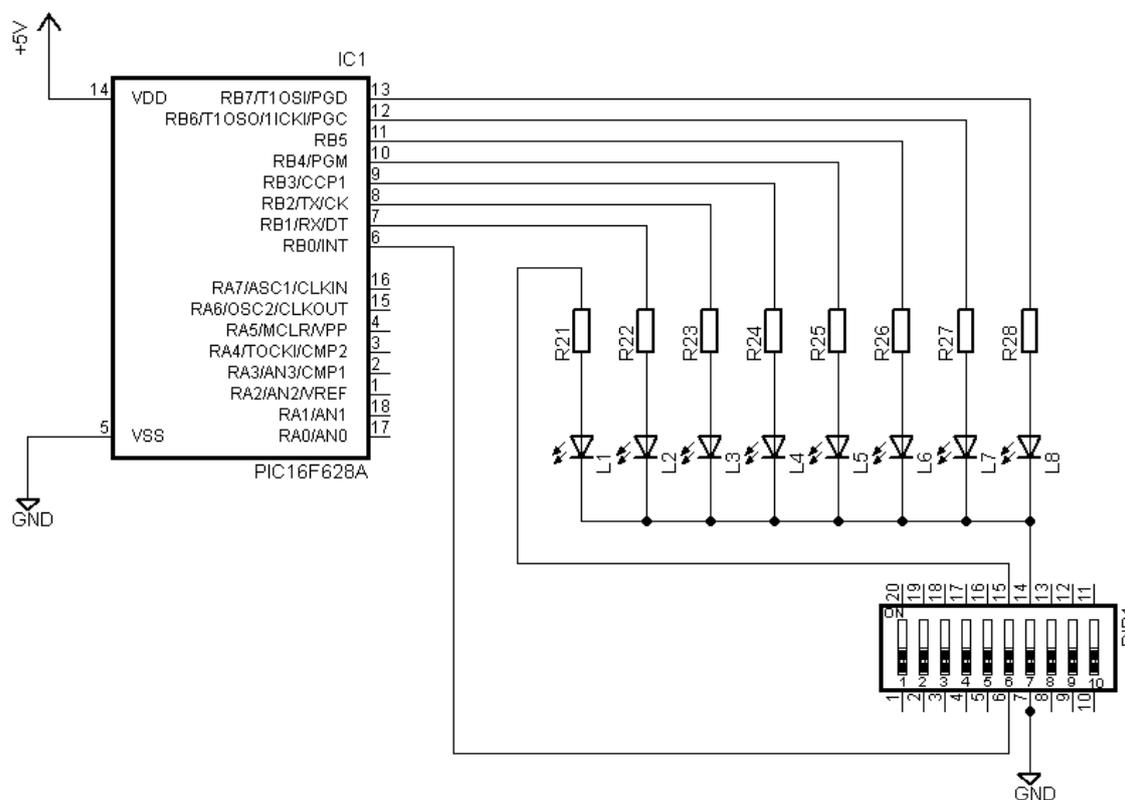


Figura 3.17 – Descrição do circuito de acionamento dos LEDs

Fonte: [Manual ACEPIC 18]

O sensor infravermelho utilizado neste projeto é ligado ao kit através da porta de expansão conectada a portA RA1 do microcontrolador. Os LEDs do semáforo de veículos são ligados aos pinos RB0, RB1 e RB2, na PortB. Já os LEDs do semáforo de

pedestres foram ligados aos pinos RA3 e RA4, portA. A figura 3.18 a seguir, retirado do manual do Kit, ilustra a posição que os mesmos foram encaixados:

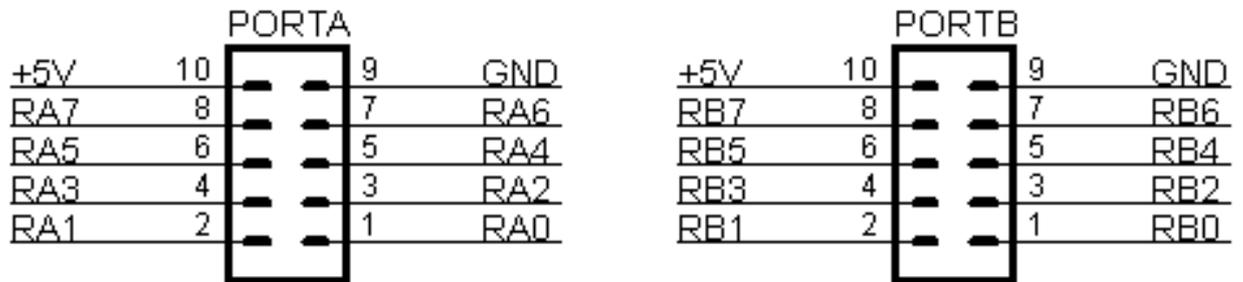


Figura 3.18– Descrição da porta de expansão

Fonte: [Manual ACEPIC 18]

CAPÍTULO 4: TESTES E IMPLEMENTAÇÃO

Neste capítulo é apresentada a descrição do planejamento para a montagem do protótipo, bem como a lógica do algoritmo e criação do software necessário para fazer o sincronismo do sinal de trânsito com o sinal de pedestres em um semáforo. No código, também foi incluído a lógica de sincronismo do sensor infravermelho, necessário para identificação de pedestres no sinal de trânsito, além do sinal sonoro (bip) utilizado para alertar os deficientes visuais ou pedestres desatentos.

4.1 PLANEJAMENTO

Para o desenvolvimento deste projeto, foi necessário planejar como implementar a solução de sincronismo. Além da consulta em bibliografias, foi também realizado um monitoramento de alguns semáforos para verificar o seu comportamento quanto ao tempo de verde, amarelo e vermelho. Em tal verificação, foram realizados todos os testes possíveis, levando em consideração que os tempos se alteram quando o botão verde é pressionado.

4.2 PLANO DE TRÁFEGO DO PROJETO

4.2.1 TESTES REALIZADOS NOS SEMÁFOROS DAS COMERCIAIS DA 203 NORTE, 303 NORTE E 114 SUL

Os testes citados abaixo foram realizados buscando entender a lógica de tempo utilizada nos semáforos das entrequadras do plano piloto, em específico o da 203 norte, 303 norte e 114 sul, semáforos que nortearam essa pesquisa. Em todos os semáforos, foram obtidos exatamente os mesmos resultados, observando que os sinais das entrequadras do plano piloto seguem a mesma lógica. Os resultados estão descritos abaixo:

Sem apertar o botão verde de pedestres, foi verificado que o sinal ficou aberto para trânsito de veículos por aproximadamente 5 minutos. Passado esse tempo, a luz amarela do sinal ficou acesa por 5 segundos e o sinal ficou fechado por 15 segundos apenas, liberando novamente o trânsito para veículos logo em seguida.

Apertando o botão verde logo após o sinal de trânsito de veículos ficar liberado (luz verde), foi verificado que o sinal demorou 2 minutos para migrar para o estado de

atenção (luz amarela). Tal luz ficou acesa por 5 segundos e, logo em seguida, o sinal ficou fechado para veículos, luz vermelha acesa, por 15 segundos.

Foi realizado um novo teste, agora apertando o botão verde de passagem de pedestres um minuto após o sinal de trânsito de veículos estar aberto, e foi verificado que o sinal demorou um minuto para mudar para o estado de atenção, no qual ficou por 5 segundos antes de fechar. O sinal permaneceu fechado por 15 segundos.

Um ultimo teste foi realizado, agora apertando o botão de transição de pedestres dois minutos após o trânsito de veículos ser liberado, sinal verde, e foi verificado que imediatamente após pressionar o mesmo, o semáforo de veículos mudou para o estado de atenção, representado pela luz amarela, no qual ficou acesa por 5 segundos. Decorrido esse tempo, o sinal mudou para a cor vermelha, ficando nesse estado por 15 segundos.

Feito todos esses testes, foi possível concluir que o sinal de veículos tem um tempo mínimo de permanência em estado aberto, representado pela cor verde. Foi notado também que, independentemente de apertar o botão de passagem de pedestres ou não, o tempo de permanência do sinal em amarelo e vermelho não foi alterado. Todos os testes citados acima foram feitos em dia de semana, em horários variados nos turnos da manhã, da tarde e da noite.

Com os dados acima, tendo como base o sinal de veículos, podemos observar que:

- Tempo de Verde: Mínimo de 120 segundos; Máximo de 300 segundos;
- Tempo de Amarelo: Obrigatoriamente 5 segundos.
- Tempo de Vermelho: Obrigatoriamente 15 segundos;

Os tempos citados acima foram feitos nas comerciais do plano piloto, cuja largura da via é de aproximadamente 10 metros, padrão em todas quadras. Segundo o DETRAN-DF, quinze segundos é tempo o suficiente para atravessar tal via, uma vez que os pedestres teriam que atravessar aproximadamente 67cm por segundo. Para vias de maior largura, o tempo que o sinal fica aberto para pedestres é ampliado, porém, não segue uma lógica padrão com relação ao tempo x espaço em segundos x centímetros, respectivamente.

4.2.2 PLANO DE TRÁFEGO

Para o desenvolvimento deste projeto e para fins de demonstração, foi criado um plano de tráfego real, caso o projeto fosse implementado em semáforos de comerciais do plano piloto, sempre utilizando como referência a pesquisa feita no planejamento, priorizando a eficácia do trânsito e, ao mesmo tempo, a segurança na travessia dos pedestres. O plano tem o tempo total do ciclo de, no mínimo, 145 segundos e este ciclo está dividido da seguinte maneira, tendo como base o sinal de veículos:

- Tempo de Verde: Mínimo de 120 segundos; Máximo indeterminado;
- Tempo de Amarelo: Obrigatoriamente 5 segundos.
- Tempo de Vermelho: Obrigatoriamente 20 segundos;

Unicamente para fins de demonstração na defesa do projeto, foi criado o seguinte plano de tráfego fictício de, no mínimo, 33 segundos. Tal plano está explicitado logo abaixo, tendo como base o sinal de veículos:

- Tempo de Verde: Mínimo de 20 segundos; Máximo indeterminado;
- Tempo de Amarelo: Obrigatoriamente 3 segundos.
- Tempo de Vermelho: Obrigatoriamente 10 segundos;

4.3 ALGORITMO

O algoritmo foi desenvolvido após a obtenção de todos os dados necessários, referentes aos *hardwares* usados no projeto, após a conclusão do planejamento do projeto e com o plano de tráfego em mãos, onde foi possível montar o mapa com o acionamento das luzes indicadoras do semáforo e seus respectivos tempos.

Basicamente, a codificação está dividida da seguinte maneira:

- Início: Zera todas as variáveis utilizadas na programação, ajusta os timer's que serão utilizados e chama as rotinas para iniciar o programa.

- Loop: Faz com que o programa percorra toda a tabela com o plano de tráfego.
- Delay: Utilizado para gerar um tempo de aproximadamente 0,1 segundo.
- Tabela de planos: Tabela que contém as informações de sequenciamento das cores do semáforo que serão acionadas conforme a variação do tempo.
- Temporização: Gera o tempo necessário de acordo com cada fluxo, podendo variar de 5 a 20 segundos, dependendo do fluxo em questão.

A programação do PIC foi realizada em linguagem de programação Assembly e do tipo RISC (Computador com conjunto reduzido de instruções), com cerca de 35 instruções. A programação foi feita por meio do software “MPLAB” e o sincronismo com o microcontrolador foi feito pelo software ‘WINPIC’, que utiliza o arquivo hexadecimal para programar o PIC, gerado pelo software de programação e compilado através da porta serial.

Inicialmente, quando o programa é executado, ele mantém o semáforo de veículos aberto, sinalizado pela luz verde, e o semáforo de pedestre fechado, luz vermelha. Depois de 20 segundos, tempo obrigatório de permanência do semáforo de veículos aberto, ele inicia o processo de sincronismo e obtém as informações do acionamento ou não dos sensores de presença de pedestres.

Na tabela com o plano de tráfego estão as informações de acionamento das luzes do semáforo bem como o tempo em que esta configuração deverá se manter acesa. Quanto este tempo finaliza, o programa checa as informações do sensor de presença e, caso haja pedestres querendo atravessar a rua, há a mudança de estado do sinal de trânsito e, conseqüentemente, a mudança do sinal de pedestres, por tempos pré determinados, retornando para o fluxo inicial de plano de tráfego.

4.4 SOFTWARE

A primeira instrução do software é utilizada para identificar se os pinos do sistema serão utilizados como entrada ou saída de dados. Para tal, cada registrador configura um pino como entrada digital se o bit respectivo possuir valor “1”; ou saída

digital, se o bit respectivo possuir o valor “0”. As linhas citadas abaixo mostram os valores que devem estar presentes nos registradores TRISA e TRISB para configuração dos pinos deste projeto:

```
mask_trisa equ    b'00111111' ;Máscara para configurar os pinos do portA
mask_trisb equ    b'00000000' ;Máscara para configurar os pinos do portB
```

A quantidade máxima de sequências que tal projeto terá serão três, visto que o semáforo de veículos possui apenas 3 fases (verde, amarelo e vermelho) . Como citado anteriormente, a temporização do protótipo ficou definida em vinte segundos no estado de ‘Liberado’, cinco segundos em ‘Atenção’ e quinze segundos em ‘Fechado’, tendo como base o semáforo de veículos. Tais tempos também foram definidos nessa etapa da codificação. Após definido a primeira parte organizacional do projeto, a parte inicial do código ficou implementada da seguinte forma:

```
; DESCRIÇÃO GERAL
; Ao iniciar, o software aguardará o comando de start. Após o start teremos a seguinte
; sequencia:
; 01=> Transito de veiculos liberado; Transito de pedestres bloqueado
; 02=> Transito de veiculos em atenção; Transito de pedestres bloqueado
; 03=> Transito de veiculos bloqueado; Transito de pedestres liberado
;
;(1)Semáforo de Veículos aberto, pedestres bloqueados por 20s
;(2)Semáforo de veículos em atenção, pedestres bloqueados por 5s
;(3)Semáforo de veículos fechado, semáforo de pedestres aberto por 10s
;
;Entradas (pull-down)
; RA1 <= Sensor de presença de pedestre
;
;Saídas
; RA3 => Lâmpada verde pedestre
; RA2 => Lâmpada vermelho pedestre
; RB0 => Lâmpada verde farol
; RB1 => Lâmpada amarelo farol
; RB2 => Lâmpada vermelho farol
;-----
#include <P16F628A.INC>
;-----
;
; DEFINIÇÕES
;
mask_trisa equ b'11000111' ;mascara para configurar os pinos do portA
mask_trisb equ b'00000000' ;mascara para configurar os pinos do portB
max_sequencia equ .3 ;qtdade máxima de sequência
;
; Mascaras p/ sequência de acendimentos das lâmpadas dos semáforos:
mask_portb01 equ b'10000001' ; Avenida liberada (0000.0001)
mask_portb02 equ b'10000010' ; Avenida atenção (0000.0010)
```

```

mask_portb03 equ b'10000100' ; Avenida bloqueada (0000.0100)
;
mask_porta01 equ b'10010000' ; Pedestres bloqueados (0001.0000)
mask_porta02 equ b'10010000' ; Pedestres bloqueados (0001.0000)
mask_porta03 equ b'10001000' ; Pedestres liberados (0000.1000)
;
; TEMPORIZAÇÃO DO SEMÁFORO
tempo1 equ .200
tempo2 equ .50
tempo3 equ .100

```

Ao iniciar o programa, o código deve configurar o PIC, de modo que o que foi definido na parte inicial da programação seja inserido no microcontrolador. Para isso, o código ficou definido da seguinte maneira:

```

; CONFIG
;
Config_Pic
movlw .7           ; Move um número de valor “k” para “W” - Desliga os comparadores
                   ; dos pinos ra0, ra1 e ra2
movwf CMCON       ; Move um número de valor “w” para “F”
Bank1_on
movlw mask_trisa  ; Move um número de valor “k” para “W” - Mascara p/ configurar os
                   ; pinos do portA
movwf TRISA       ; Move um número de valor “w” para “F”
movlw mask_trisb  ; Move um número de valor “k” para “W” - Mascara p/ configurar os
                   ; pinos do portB
movwf TRISB       ; Move um número de valor “w” para “F”
Bank0_on
Return

```

Já para configurar os hardwares presentes na placa do kit, foi necessário usar a seguinte linha de comando:

```

__CONFIG _CP_OFF & _LVP_OFF & _BOREN_OFF & _MCLRE_ON &
_WDT_OFF & _PWRTE_ON & _INTRC_OSC_NOCLKOUT

```

Cada código citado acima tem uma função específica para o microcontrolador que podem ser acionadas ou desligadas, dependendo do projeto em questão. As funções inseridas neste projeto possuem as seguintes características:

- O MCLRE (Master Clear Enable), se estiver ativado, reserva o pino MCLR para a função de reset do microcontrolador.
- O LVP (Low Voltage Program) é um recurso que permite que o microcontrolador seja gravado sem a necessidade de aplicar uma tensão de cerca de 13V no pino VPP

- O BOREN (Brown-out reset) é um recurso que monitora a tensão de alimentação e quando ela cai abaixo de 4,5V provoca o reset.
- O WDT (Watch Dog Timer) é um recurso que reinicia o microcontrolador, caso o programa travar.
- O _INTRC_OSC_NOCLKOUT é o oscilador interno do kit.
- CP é um recurso para proteger a memória de programa contra cópias. Esses recursos interessam a quem fabrica aparelhos eletrônicos e quer evitar engenharia reversa.

O pino TRISA é responsável por todos os pinos presentes na PORTA. Já o TRISB é responsável por todos os pinos na PORTB. Movendo a máscara do TRISA, como explícito no código, faz com que o microcontrolador busque na parte inicial do programa a primeira linha da parte ‘DEFINIÇÕES’, referente a máscara para configurar os pinos PORTA. O mesmo acontece logo abaixo com PORTB, através da linha de comando referente ao TRISB. Bank0 e Bank1 são labels referentes aos bancos da memória aonde o programa será executado. Após tais definições e comandos, o sistema retorna para o fluxo principal, através do comando ‘Return’.

O fluxo principal do programa representa a lógica central do mesmo e o funcionamento correto do sistema depende desta.

```

;-----
;INICIO DO PROGRAMA
;-----
Inicio
call Config_Pic           ; configuração dos ports
clr PORTA                ; Limpa o conteudo de f
clr PORTB                 ; Limpa o conteudo de f
Loop
clr wsequencia           ; Limpa o conteudo de f
call LigaLampadasDoFarol ; Chama a subrotina
call Delay                ; Chama a subrotina
incf wsequencia,f        ; Incrementa apontando p/ próxima sequência
                          ; fica nesta situação até que um pedestre, esteja
                          ; presente no semáforo

ChkSensoresPresenca
movlw b'00000001'        ; Move o numero de valor k para w
andwf PORTA,W            ; Operacao logica 'E' entre os dois valores
btfsc flag_zero          ; Verifica o valor do bit b do registrador f e pula a
                          ; proxima instrucao se for igual a 0 - Tem carro
                          ; ou pedestre na av.

goto ChkSensoresPresenca ; Vai para o endereco k se nao tiver pedestre
Loop2                     ; Vai para Loop2 se tiver pedestre
call LigaLampadasDoFarol ; Chama a subrotina
call Delay                ; Chama a subrotina
incf wsequencia,f        ; Incrementa apontando p/ próxima sequência

```

```

movlw max_sequencia           ; Move o numero de valor k para w – Quantidade
                               ; máxima de execução
xorwf wsequencia,w           ; compara wsequencia com 3
btfss flag_zero              ; Verifica o valor do bit “b” do registrador “f” e pula
                               ; a próxima instrução se for igual a “1” - Fim de
                               ; ciclo
goto Loop2                   ; Vai para Loop 2 se nao for o fim do ciclo
goto Loop                    ; Vai para Loop se for o fim do ciclo

```

É indispensável que a memória do microcontrolador esteja posicionada para a execução da primeira linha do fluxo principal logo ao dar início ao processamento do código. Tal comando é definido da seguinte forma:

```

; DECLARAÇÃO DE VARIÁVEIS DE RAM
;
cblock 0X20                   ; Endereço inicial da RAM
wdg0_d101                    ; Digito - significativo delay_01
wdg1_d101                    ; Digito + significativo delay_01
wtimer                       ; Base de tempo, em segundos, p/ rotina delays
wsequencia
endc

```

A expressão 0X20 após a diretiva CBLOCK indica o endereço na memória de dados que assumirá a primeira variável definida. Após tal definição, foi utilizado o programa MPLAB para verificar se o programa está apontando para o lugar correto. Na figura a seguir, é possível observar que o sistema está apontando para a primeira linha da programação, que está indicada pela seta verde na figura 4.1.

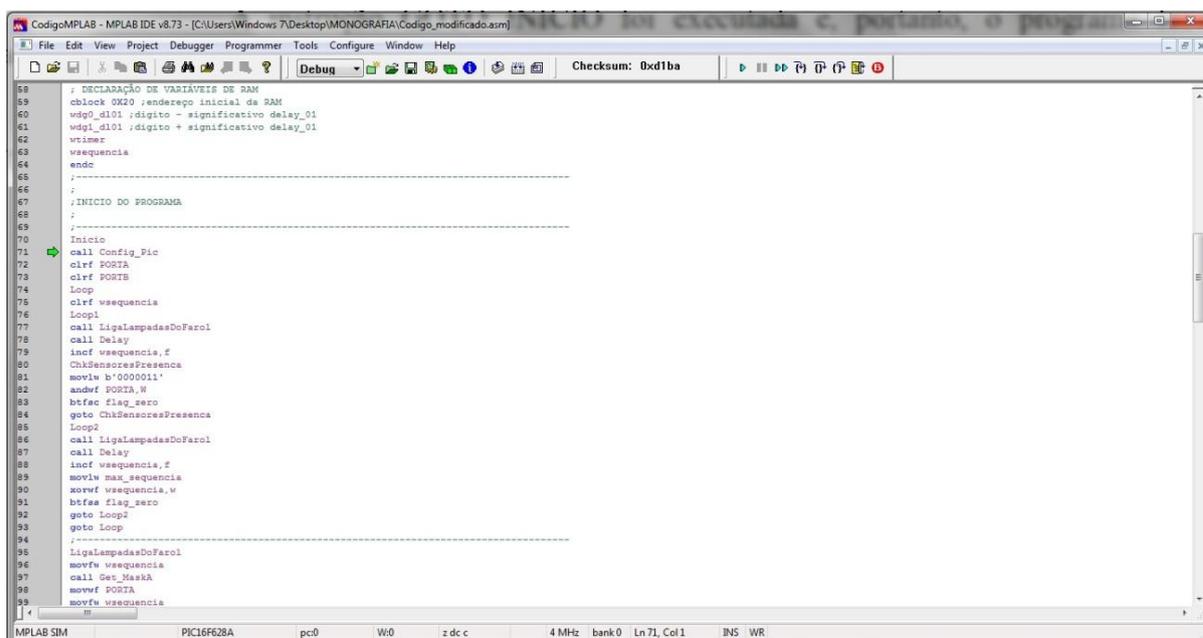


Figura 4.1 – Início da execução do programa

Fonte: Autor

Após limpar o conteúdo da PORTA, PORTB e da sequência, o programa aciona a linha ‘LigaLampadasDoFarol’. Tal execução será perceptível, pois irá ligar os LEDs que representam os sinais de trânsito de veículo e pedestres.

```

LigaLampadasDoFarol
movfw wsequencia      ; Move o conteúdo de F para W
call Get_MaskA        ; Chama a subrotina - Pega a mascara p/ ligar as lâmpadas do
                      ; Semáforo de veículos
movwf PORTA           ; Move o conteúdo de w para f - Farol conectados ao portA
movfw wsequencia      ; Move o conteúdo de F para W
call Get_MaskB        ; Chama a subrotina - Pega a mascara p/ ligar as lâmpadas do
                      ; semáforo de pedestres
movwf PORTB           ; Move o conteúdo de w para f - Farol conectados ao portB
return

```

Tal função irá pegar a sequência de bits primeiramente da PORTA e substituir pela máscara contida na primeira linha da parte de ‘DEFINIÇÕES’, ou seja, irá ligar o LED vermelho do semáforo de pedestres. O mesmo irá executar a mesma ação na PORTB, ligando o LED verde do semáforo de veículos. Feito isso, o sistema retorna para o fluxo principal, apto para realizar a próxima ação.

Após as máscaras serem devidamente selecionadas e executadas, o sistema chama a subrotina ‘Delay’, que irá determinar quantos segundos o sistema permanecerá nessa definição. Porém, a função delay apenas causa um atraso no sistema de 0,1 segundo. O que determina o tempo em que o sinal permanecerá em cada uma das fases foi determinado no início do programa, nas linhas ‘tempo1’, ‘tempo2’ e ‘tempo3’. Tais funções são chamadas através do comando:

```

; GET_PVTIMER
;
; Recebe o número da sequência do semáforo em wsequencia e retorna com o valor do preset
; do timer p/ o delay
;
Get_PvTimer
addwf PCL,f           ;Soma o conteúdo do registrador 'W' e de 'f'
retlw tempo1         ;Retorna com um valor 'k' no registrador 'W'
retlw tempo2         ;Retorna com um valor 'k' no registrador 'W'
retlw tempo3         ;Retorna com um valor 'k' no registrador 'W'

```

Tal comando encerra o fluxo principal do primeiro ciclo. Após isso, os contadores são incrementados para o início do segundo ciclo, que acende a luz amarela do semáforo de veículos. Neste, o semáforo de pedestres continua fechado, visto que o trânsito de veículos ainda não foi interrompido. Feito isso, dá-se início ao terceiro e

ultimo ciclo, aonde o sinal de veículos fecha, liberando a passagem segura dos pedestres.

Terminado os três ciclos, o sistema zera o contador de sequência, e permanece no estado de espera com o sensor de pedestres ligado. Este, ao ser acionado com a interrupção do feixe infravermelho, incrementa o 'wsequencia', dando início novamente aos três ciclos.

4.5 MONTAGEM DO PROTÓTIPO

Para a confecção deste projeto, foi construído um protótipo para simular a situação real de um pedestre querendo atravessar a faixa de pedestres em semáforos do plano piloto, foco principal deste projeto. Os postes com sensor infravermelho, que o pedestre terá que cruzar para acionar o sistema, são reproduzidos por dois canos de PVC de aproximadamente um metro de altura e as luzes que serão acionadas para representar os semáforos serão simuladas por LEDs de alto brilho, que serão acopladas em um protótipo imitando um sinal de trânsito real.

Os primeiros testes para a confecção deste projeto ocorreram em um ambiente virtual, utilizando um software de simulação de circuitos elétricos chamado Proteus, versão 7.1 da empresa Labcenter Electronics, por indicação da professora orientadora.

Os primeiros testes com o software não foram satisfatórios por ausência de linhas de código obrigatórias para o funcionamento do sistema. Depois de resolvidos esses problemas, o protótipo foi montado da maneira adequada e funcionou como esperado. Porém, nessa primeira parte de montagem, o protótipo foi montado apenas em um protoboard, para simular o funcionamento correto de todo sistema. A figura a seguir ilustra o protoboard utilizado para confecção da parte de comunicação dos *hardwares* do projeto com o KIT do microcontrolador.

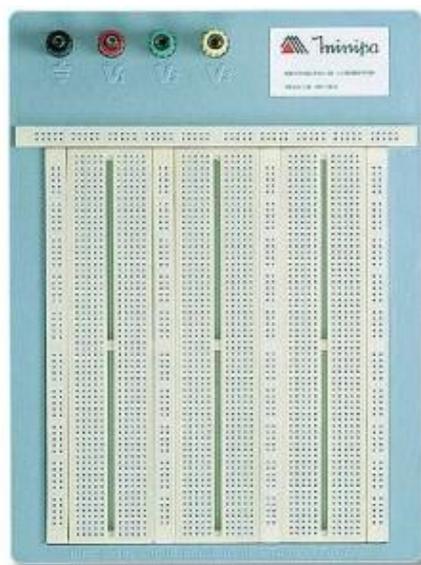


Figura 4.2 – Protoboard utilizado para montagem do sistema

Fonte: www.minipa.com.br

As versões do software foram testadas tanto no simulador, como no kit do PIC 16F628A da ACEPIC, citado anteriormente, adquirido para confecção do protótipo. Para testar o software no kit, sem a instalação do sensor, foram utilizados os botões de acionamento de interrupções, presentes na placa controladora do kit, e as luzes dos semáforos foram representadas pelos LEDs do próprio kit.

Depois de finalizado o software, foi montado no *protoboard*, com ajuda da professora orientadora, um circuito para representar o semáforo de trânsito de veículos através de três LEDs, verde, amarelo e vermelho. Além disso, foi simulado também o semáforo de pedestres, através de dois LEDs: Verde e vermelho. Para finalizar, foi montado o circuito no sensor infravermelho, através de um emissor e um receptor em linha, ambos de baixo custo, apenas para caráter experimental. As figuras 4.3 e 4.4 a seguir ilustram tal montagem.

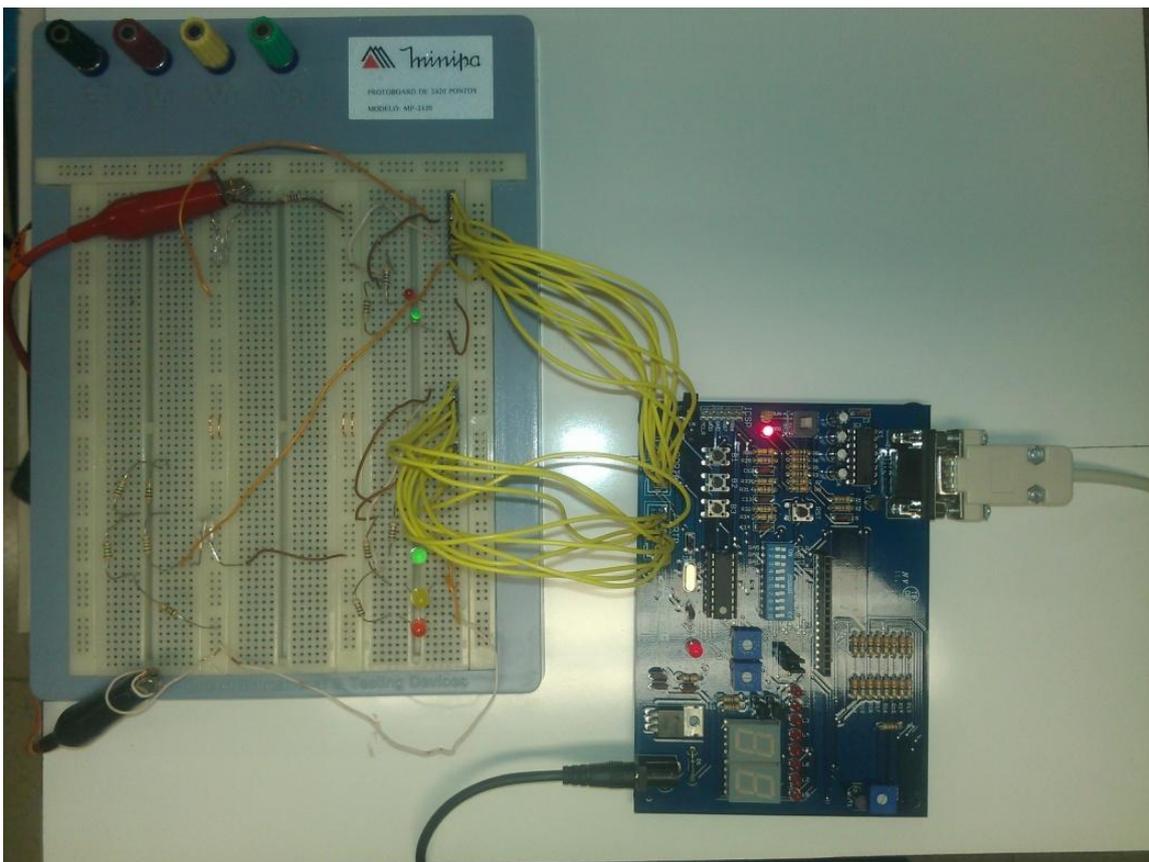


Figura 4.3 – Protótipo inicial montado

Fonte: Autor



Figura 4.4 – Simulação do semáforo de veículos

Fonte: Autor

Na figura 4.4, é possível notar a presença de resistores em série com cada um dos LEDs do protótipo. Estes, foram colocados para evitar qualquer tipo de passagem de corrente, maior ou menor ao esperado. Os resistores usados são de apenas $100\ \Omega$, mais que suficiente para evitar qualquer tipo de passagem de corrente incorreta. Esses

foram instalados em série com os LEDs, seguindo a linha de tensão lógica do *protoboard*. Os fios amarelos estão conectando os LEDs aos seus respectivos pinos, no KIT. As luzes verde, amarela e vermelha, do semáforo de veículos, estão ligadas, respectivamente, aos pinos RB0, RB1 e RB2, da PORTB do KIT. Já as luzes verde e vermelha do semáforo de pedestres estão ligadas as portas RA3 e RA2 da PORTA do KIT, respectivamente.

Para confecção do sensor infravermelho emissor receptor, foi utilizado um fototransistor e um receptor, ambos de baixo custo. A figura a seguir ilustra tal função.

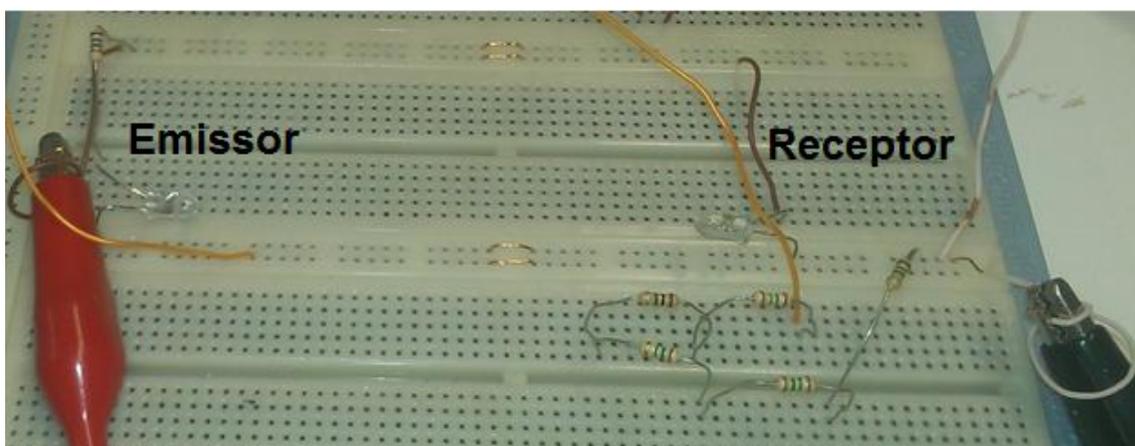


Figura 4.5 – Circuito do sensor

Fonte: Autor

Na figura 4.5 ficou claro também a presença de uma série de resistores. Após pesquisa, estudo e alguns testes junto com o professor orientador, foi notado que quanto maior a carga no receptor e menor a carga no emissor, maior ficava a sensibilidade e eficiência do sistema, possibilitando a detecção em um maior intervalo. Após uma série de simulações, ficou definido que o emissor necessitava de quatro resistores de $1.5\text{ k}\Omega$, totalizando $6\text{ k}\Omega$, para seu perfeito funcionamento no espaço desejado, que será de aproximadamente um metro. Já o receptor recebeu um resistor de $200\ \Omega$, mais que suficiente para identificar $4,5\text{V}$ no intervalo presente no protoboard e aproximadamente $2,5\text{V}$ na distância de aproximadamente um setenta centímetros presente no protótipo que será apresentado a banca. Essa tensão é identificada como nível lógico 1 pelo microcontrolador. Ao interromper o feixe de luz infravermelha, a voltagem cai para 0V ,

sendo identificado como nível lógico 0 pelo microcontrolador. Tal sensor foi conectado a entrada RA0 da portA do KIT.

Após a montagem do protótipo inicial para verificação dos hardwares e software do projeto, foi dado início a montagem do protótipo final, para demonstração a banca. Para a montagem dos semáforos, foram utilizados isopor, tinta preta, funis da cor branca, plástico verde, amarelo e vermelho e fios de cobre.

Primeiramente, o isopor foi recortado para ficar no tamanho aproximado de um sinal de trânsito real, furado para a inserção dos funis e pintado da cor preta. Feito isso, os LED's foram inseridos e o papel plástico foi colado ao funil, seguindo a cor do LED. A figura 4.6 a seguir ilustra a parte de trás do semáforo, aonde os LED's foram inseridos.

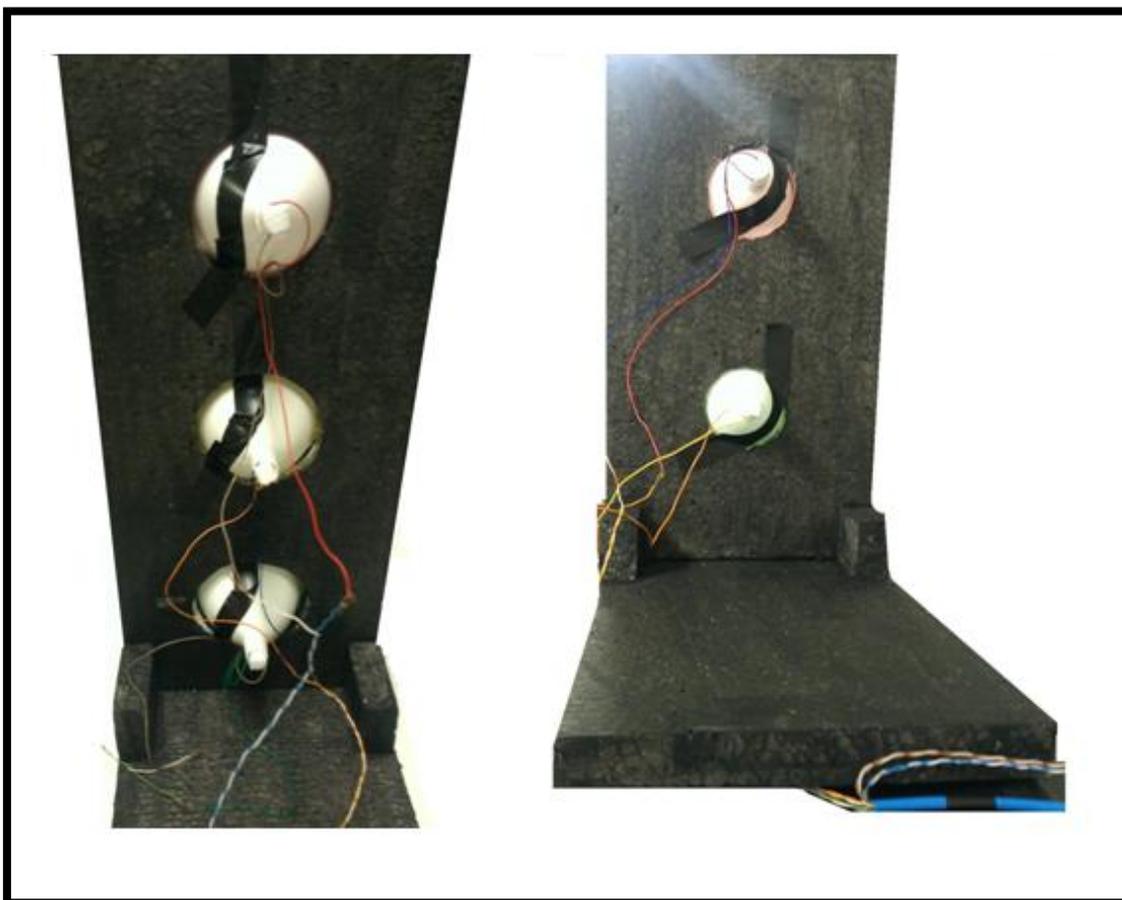


Figura 4.6 – Montagem dos semáforos

Fonte: Autor

A figura 4.7 ilustra os semáforos já finalizados:

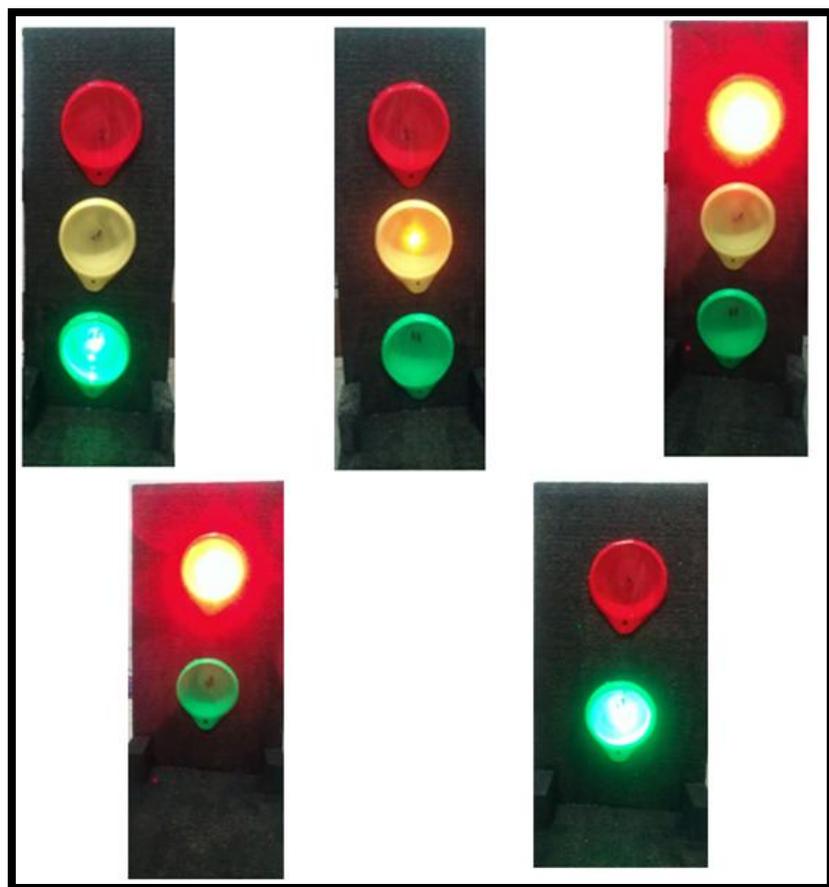


Figura 4.7 – Semáforos finalizados

Fonte: Autor

Após finalizar a montagem completa do semáforo, deu-se início a montagem dos sensores para demonstração a banca. Na montagem, além do emissor e do receptor, foram utilizados dois canos perfeitamente alinhados, montados em uma base de plástico. Para conexão dos sensores ao protoboard, foram utilizados fios de impressora. A figura 4.8 ilustra o sistema de detecção de pedestres finalizado.



Figura 4.8 – Sensor de presença finalizado

Fonte: Autor

Após finalizadas as montagens dos semáforos e do sensor de presença, o protótipo foi dado como concluído. A figura 4.9 ilustra o protótipo apresentado a banca.

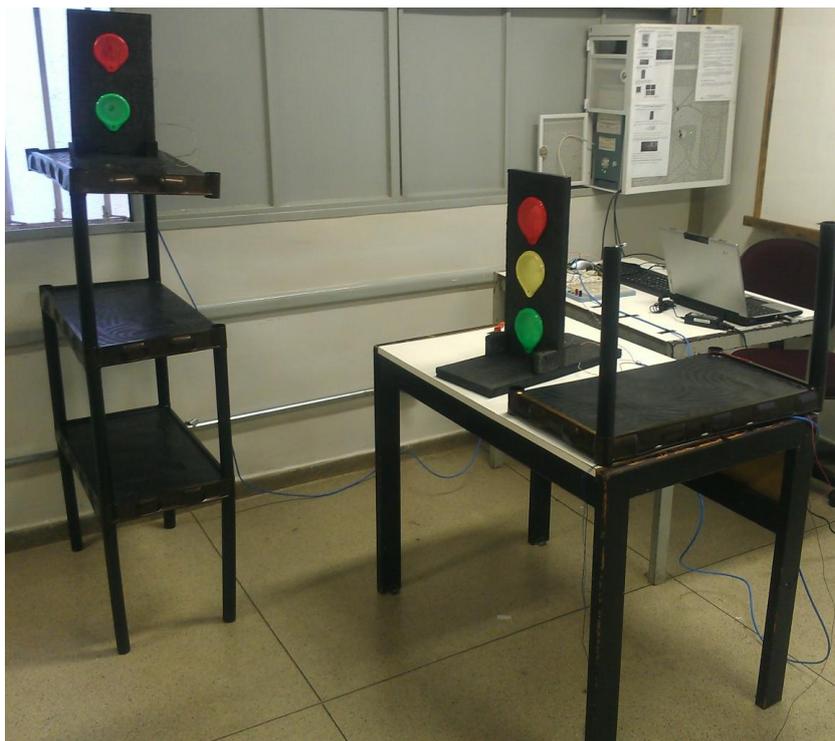


Figura 4.9 – Protótipo finalizado

Fonte: Autor

4.6 ORÇAMENTO

Para a realização deste projeto, foram efetuados os seguintes gastos:

Item	Quantidade	Custo Unitário	Custo Final
KIT Acepic 16F628A	1	R\$ 192,50	R\$ 192,50
LED Alto Brilho	5	R\$ 1,50	R\$ 7,50
Buzzer	1	R\$ 5,00	R\$ 5,00
Fio de impressora	2m	R\$ 4,00	R\$ 8,00
Resistores	8	R\$ 1,00	R\$ 8,00
Fototransistor	1	R\$ 2,00	R\$ 2,00
Receptor	1	R\$ 2,00	R\$ 2,00
Conector	30	R\$ 0,30	R\$ 9,00
Isopor	2	R\$ 2,50	R\$ 5,00
Cola de Isopor	1	R\$ 3,50	R\$ 3,50
Tinta Preta	1	R\$ 5,00	R\$ 5,00
Funil	5	R\$ 2,50	R\$ 12,50
Plástico Colorido	3m	R\$ 1,50	R\$ 4,50
TOTAL	-	R\$ 223,30	R\$ 264,50

Quadro 4.1 – Custos do Projeto

Fonte: Autor

CAPÍTULO 5: CONCLUSÃO

O projeto efetivamente atingiu o seu objetivo principal, que é de proporcionar segurança e praticidade aos pedestres usuários das faixas de pedestres em comerciais do plano piloto. Durante a implementação deste projeto, surgiram vários problemas, principalmente com relação ao software implementado no sistema. Tais problemas ocorreram devido a falta de conhecimento pleno em tal linguagem e serviram como aprendizado a uma linguagem de programação antes desconhecida e crescimento intelectual que serão de grande valia na minha vida profissional.

A falta de manuais e explicação muito sucinta nos documentos encontrados referente ao kit da ACEPIC fez com que o tempo gasto para fazer uma simples leitura e envio de dados ao microcontrolador fosse muito superior ao planejado, causando uma preocupação totalmente desnecessária se os manuais fossem devidamente elaborados. Para resolução de tal problema, foram necessárias horas de testes, pesquisa e contato com os desenvolvedores do kit em questão.

Após a resolução de tais problemas e após uma série de testes, inclusive testes de estresse e de condições adversas de luminosidade do local, foi constatado que todos os objetivos citados na introdução da monografia foram atingidos com sucesso, sendo demonstrado então que é possível produzir tal projeto em escala industrial e implementar em todas as comerciais do plano piloto e entorno, que possuam via de mão dupla sem cruzamentos.

O projeto conseguiu alcançar o objetivo de aumentar a segurança nas comerciais do plano piloto, por obrigar os pedestres a se dirigirem ao semáforo, uma vez que este não irá fechar se não houver pessoas querendo atravessar a rua. Tal fato também possibilita a implementação de radares de trânsito nas comerciais do plano piloto, já que o avanço do sinal vermelho poderia causar atropelamento de pedestres e/ou animais.

O projeto se mostrou moderno e de baixo custo, sendo possível sua implementação em outras vias aonde a largura e o plano de tráfego sejam diferentes, pois a temporização do projeto pode ser alterada em seu programa, seguindo a legislação de trânsito do local em questão.

Tal projeto também conseguiu atingir seu principal e mais importante objetivo: a priorização de cadeirantes e deficientes visuais em um simples ato de ir e vir de forma segura, mais rápida e eficiente. Com o sistema antigo, tais pessoas não conseguiam usufruir de tal benefício, devido ao difícil acesso e a não priorização de tal função a essas pessoas que, talvez, sejam as que mais precisem do mesmo devido suas limitações.

5.1 SUGESTÕES DE TRABALHOS FUTUROS

Como sugestão para outros projetos nesta área, recomenda-se o controle também da quantidade de veículos presentes no trânsito, fazendo uma comparação com a quantidade de pedestres e liberando ou não a passagem dos mesmos, de acordo com os resultados obtidos. Recomenda-se ainda, implementar tal função em locais aonde há cruzamento, como na via W3, caso o projeto seja baseado no trânsito do Distrito Federal. É possível também, incluir um monitor LCD que indicaria quando o semáforo já tiver atingido seu tempo obrigatório na primeira sequência (120 segundos), mostrando ao usuário presente que se o mesmo interromper o feixe de luz infravermelho naquele instante, o sinal mudará para a sequência seguinte. Por fim, é possível adicionar códigos relativos a banco de dados, fazendo um gráfico comparativo do número de pessoas e carros que passam por aquele semáforo em determinados horários do dia ou em determinados dias da semana.

REFERÊNCIAS BIBLIOGRÁFICAS

ACEPIC, (2011) – Datasheet ACEPIC_18.

Disponível em: www.acepic.com.br/manuais/acepic_18.pdf

Acesso em: 10/03/2012.

CIRCUIT SHOP. (2011) – MAX232.

Disponível em: http://circuitshop.com.br/loja/popup_image.php?PID=22

Clube do Hardware – Tutorial de programação assembly para microcontroladores PIC.pdf

Disponível em: <http://forum.clubedohardware.com.br/eletronica/f39>

Acesso em: 25/05/2012

FOROUZAN, Behrouz. – Comunicação de dados e redes de computadores.

3a edição. Editora Bookman. São Paulo 2008.

Acesso em: 12/04/2012.

JOSÉ, A. (2009) – *Microcontroladores*. Serpa, Portugal.

Disponível em: <http://lusorobotica.com/index.php?topic=1198.0>.

Acesso em: 26/04/2012.

KIOSKEA, (2011) – Imagem conector DB9.

Disponível em: <http://pt.kioskea.net>

Acesso em: 12/05/2012.

LABTOOLS. FAQ.

Disponível em: http://www.mosaico-eng.com.br/index.asp?area=04&subarea=b&idioma=por&script=area04_02

Acesso em: 25/04/2012.

MIGUEL, Afonso – Sensor Infravermelho.

Disponível em: <http://www.afonsomiguel.com/content/sensor-infra-vermelho>

Acesso em: 25/10/2011.

MARKUS, Otávio. – Circuitos Elétricos.

8ª edição. Editora Érica.

RADCOMALARMES, (2011) – Imagem infravermelho detecção reflexão.

Disponível em: <http://www.radcomalarmes.com/about-us.php>

Acesso em: 15/05/2012.

REDETEC, (2011) – Imagem infravermelho interrupção de feixe.

Disponível em: <http://www.redetec.org.br/inventabrasil/feixe.htm>

Acesso em: 15/05/2012.

SCHOR (1999) – Revolução Invisível.

Disponível em:

http://www.scielo.br/scielo.php?script=sci_arttext&pid=S010288391999000300014

Acesso em: 15/03/2012.

SIPITAKIAT, Arnan – Como fazer sensores, “GoGo board”,

Disponível em: <http://www.blikstein.com/gogo/documents/making%20sensors.html>

Acesso em: 12/04/2012.

SYSTEM, M. I. (2007) – *RS232 Pinouts Designation*.

Disponível em: http://www.machine-informationssystem.com/RS232_Pinouts.html

Acesso em: 21/04/2012.

TEXAS INSTRUMENTS, (2004) – *MAX232*. Datasheet do MAX232.

Disponível em: <http://pdf1.alldatasheet.com/datasheet-pdf/view/27230/TI/MAX232N.html>

Acesso em: 25/02/2012.

THOMAZINI, Daniel; BRAGA DE ALBUQUERQUE, Pedro Urbano – Sensores

Industriais. Fundamentos e Aplicações.
1º Edição. Editora Érica. São Paulo 2005.

TOCCI, Ronald J; WIDMER, Neal S. – Sistemas Digitais: Princípios e Aplicações.
8ª Edição. São Paulo: Prentice Hall, 2003
Capítulos: 5, 6 e 7

Winbond Eletronics Corp, San Jose, p. 24. ISD1420 – Circuito de gravação.
Disponível em: http://www.datasheetcatalog.org/datasheets/270/499551_DS.pdf
Acesso em: 15/04/2012.

APÊNDICE A

```
-----  
;  
;  
; AUTOMAÇÃO DE SEMÁFORO  
;  
; Projeto: Sensor de presença de pedestres em semáforos das comerciais do plano  
piloto  
; Objetivo: Controlar o tráfego de veículos e pedestres em comerciais do plano  
piloto  
; Autor: Henrique Cardoso  
; Data: 25/05/2012  
;  
-----  
; Descrição Geral  
; Ao iniciar, o software aguardará o comando de start. Após o start teremos a seguinte  
; seqüência:  
; 01=> Trânsito de veículos liberado; Trânsito de pedestres bloqueado  
; 02=> Trânsito de veículos em atenção; Trânsito de pedestres bloqueado  
; 03=> Trânsito de veículos bloqueado; Trânsito de pedestres liberado  
;  
; (1) Semáforo de veículos aberto, pedestres bloqueados por 20s  
; (2) Semáforo de veículos em atenção, pedestres bloqueados por 5s  
; (3) Semáforo de veículos fechado, semáforo de pedestres aberto por 10s  
;  
; Entradas (pull-down)  
; RA0 <= Sensor de presença de pedestre  
;  
; Saídas  
; RA3 => Lâmpada verde pedestre  
; RA2 => Lâmpada vermelho pedestre  
; RB0 => Lâmpada verde farol  
; RB1 => Lâmpada amarelo farol
```

```

; RB2 => Lâmpada vermelho farol
; RB5 => Buzzer
;-----
#include <P16F628A.INC>
;-----
; CONFIGURAÇÃO DOS BITS
;
__CONFIG _CP_OFF & _LVP_OFF & _BOREN_OFF & _MCLRE_ON &
_WDT_OFF & _PWRTE_ON & _INTRC_OSC_NOCLKOUT
;
; DEFINIÇÕES
;
mask_trisa equ b'10000001'      ; Máscara para configurar os pinos do portA
mask_trisb equ b'00000000'      ; Máscara para configurar os pinos do portB
max_sequencia equ .3            ; Quantidade máxima de sequência
;
; Máscaras para sequência de acendimentos das lâmpadas dos semáforos
;
mask_portb01 equ b'10000001'    ; Avenida liberada; buzzer inativo (0000.0001)
mask_portb02 equ b'10000010'    ; Avenida atenção; buzzer inativo (0000.0010)
mask_portb03 equ b'10100100'    ; Avenida bloqueada; buzzer ativo (0010.0100)
;
mask_porta01 equ b'10000100'    ; Pedestres bloqueados (0000.0100)
mask_porta02 equ b'10000100'    ; Pedestres bloqueados (0000.0100)
mask_porta03 equ b'10001000'    ; Pedestres liberados (0000.1000)
;
; TEMPORIZAÇÃO DO SEMÁFORO
;
tempo1 equ .200                  ; Sinal de veículos aberto – 20 segundos
tempo2 equ .50                   ; Sinal de veículos em atenção – 5 segundos
tempo3 equ .100                  ; Sinal de veículos fechado – 10 segundos
;-----
;
; DEFINES

```

```

;
#define liga bsf
#define desliga bcf
#define SensorPedestre PORTA,1
#define flag_zero STATUS,2
#define Bank1_on bsf STATUS,5 ; Muda para "1" o valor do bit 5 de STATUS
#define Bank0_on bcf STATUS,5 ; Muda pra "0" o valor do bit 5 de STATUS
;-----
;
;DECLARAÇÃO DE VARIÁVEIS DE RAM
;
        cblock 0X20                ; Endereço inicial da RAM
            wdg0_dl01                ; Digito - significativo delay_01
            wdg1_dl01                ; Digito + significativo delay_01
            wtimer                    ; Base de tempo, em segundos, p/ rotina 'delays'
            wsequencia                ; Contador de sequências (MÁX = 3)
        endc

        ORG 0X00                    ; Posição da memória aonde inicia o programa
        GOTO INICIO                  ; Vai para o inicio

        ORG 0X04                    ; Posição da memória caso o programa trave
        RETFIE                        ; Se travar, reinicia na posição 0X04 da memória
;-----
;INICIO DO PROGRAMA
;-----
INICIO
        call Config_Pic                ; Configuração das PORT's
        clrf PORTA                    ; Limpa o conteúdo de PortA
        clrf PORTB                    ; Limpa o conteúdo de PortB
Loop
        clrf wsequencia                ; Limpa o conteúdo de wsequencia
Loop1
        call LigaLampadasDoFarol        ; Chama a subrotina LigaLampadasdoFarol

```

```

call Delay ; Chama a subrotina 'Delay'
incf wsequencia,f ; Incrementa apontando p/ próxima
; sequência - Fica nesta situação até que um
; pedestre esteja presente na outra avenida

ChkSensoresPresenca
    movlw b'00000001' ; Liga o sensor de presença
    andwf PORTA,W ; Operação lógica 'E' entre os dois valores
    btfsc flag_zero ; Verifica o valor do bit b do registrador f e
; pula a próxima instrução se for igual a 0 -
; Tem pedestre no semáforo.

    goto ChkSensoresPresenca ; Volta para cima se não tiver pedestre
Loop2 ; Vai para Loop2 se tiver pedestre
    call LigaLampadasDoFarol ; Chama a subrotina LigaLampadasDoFarol
    call Delay ; Chama a subrotina Delay
    incf wsequencia,f ; Incrementa apontando p próxima sequência
    movlw max_sequencia ; Move o numero de valor k para w -
; Quantidade máxima de execução

    xorwf wsequencia,w ; Operação lógica "OU EXCLUSIVO" do
; conteúdo de "W" com o conteúdo de "f" -
; compara wsequencia com 3

    btfss flag_zero ; Verifica o valor do bit "b" do registrador
; "f" e pula a próxima instrução se for igual
; a "1" - Fim de ciclo

    goto Loop2 ; Vai para Loop 2 se não for o fim do ciclo
    goto Loop ; Vai para Loop se for o fim do ciclo
;-----
; SUBROTINAS
;-----
;
LigaLampadasDoFarol
    movfw wsequencia ; Move o conteúdo de F para W
    call Get_MaskA ; Chama a subrotina - Pega a mascara p/
; ligar as lâmpadas do Semáforo de veículos

    movwf PORTA ; Inicializa a função 'PortA' - Farol ped.

```

```

; conectados ao portA
movfw wsequencia      ; Move o conteúdo de F para W
call Get_MaskB        ; Chama a subrotina - Pega a mascara p/
;ligar as lâmpadas do semáforo de pedestres
movwf PORTB          ; Inicializa a função 'PortB' – Farol veic.
; conectados ao portB

return

;-----
;
; GET_MASKA
;
; Recebe o número da sequência do semáforo em wseq e retorna com a máscara para
; ligar as lâmpadas em w
;
Get_MaskA
    addwf PCL,f        ; Soma o conteúdo do registrador 'W' e de 'f'
    retlw mask_porta01 ; Retorna com um valor k no registrador 'W'
    retlw mask_porta02 ; Retorna com um valor k no registrador 'W'
    retlw mask_porta03 ; Retorna com um valor k no registrador 'W'
;-----
;
; GET_MASKB
;
; Recebe o número da sequência do semáforo em wseq e retorna com a máscara para
; ligar as lâmpadas em w
;
Get_MaskB
    addwf PCL,f        ; Soma o conteúdo do registrador 'W' e de 'f'
    retlw mask_portb01 ; Retorna com um valor k no registrador 'W'
    retlw mask_portb02 ; Retorna com um valor k no registrador 'W'
    retlw mask_portb03 ;Retorna com um valor k no registrador 'W'
;-----
;
; Aguarda tempo especificado de acordo com a sequência atual

```

```

;
Delay
    movfw wsequencia           ; Move o conteúdo de F para W
    call Get_PvTimer          ; Chama a subrotina - Pega o preset p/ delay
    movwf wtimer              ; Inicializa a função 'wtimer'
    call Delays                ; Chama a subrotina - Atraso de x segundos
    return

;-----
;
; GET_PVTIMER
;
; Recebe o número da sequência do semáforo em wsequencia e retorna com o valor do
; preset do timer p/ o delay
;
Get_PvTimer
    addwf PCL,f                ; Soma o conteúdo do registrador 'W' e de 'f'
    retlw tempo1              ; Retorna com um valor k no registrador 'W'
    retlw tempo2              ; Retorna com um valor k no registrador 'W'
    retlw tempo3              ; Retorna com um valor k no registrador 'W'

;-----
;
; DELAYS
;
; Aguarda x segundos e retorna - o tempo deve vir em wt0 - Tempo máximo de 25,5
; segundos
;
Delays
    call Delay_01             ; Chama a subrotina
    decfsz wtimer,f           ; Decrementa o valor armazenado em "f" e
                                ; pula a próxima instrução se for igual a "0"
    goto $-2                  ; Volta duas posições acima
    return

;-----
;

```

```

; DELAY_01
;
; Atraso de 0,1 segundo - Rotina calculada para oscilador de 4MHz
;
Delay_01
    movlw .100                ; Move um número de valor "k" para "W"
    movwf wdg1_dl01          ; Inicializa a função 'wdg1_dl01'
    movlw .250                ; Move um número de valor "k" para "W"
    movwf wdg0_dl01          ; Inicializa a função 'wdg0_dl01'
    nop                       ; Sem operação
    decfsz wdg0_dl01,f       ; Decrementa o valor armazenado em "f" e
                                ; pula a próxima instrução se for igual a "0"

    goto $-2                  ; Vai para duas linhas acima
    decfsz wdg1_dl01,f       ; Decrementa o valor armazenado em "f",
                                ; pula a próxima instrução se for igual a "0"

    goto $-6                  ; Vai para seis linhas acima
    return

;-----
;
; CONFIG
;
Config_Pic
    movlw .7                  ; Move um número de valor "k" para "W" –
                                ; Desligando os comparadores

    movwf CMCON               ; Inicializa o registrador 'CMCON'
                                ; dos pinos ra0, ra1 e ra2

    Bank1_on
    movlw mask_trisa          ; Move um número de valor "k" para "W" –
                                ; Máscara p/ configurar os pinos do portA

    movwf TRISA               ; Inicializa 'TRISA'

    movlw mask_trisb          ; Move um número de valor "k" para "W" –
                                ; Máscara p/ configurar os pinos do portB

    movwf TRISB               ; Inicializa 'TRISB'

    Bank0_on

```

Return

;------

END