



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

SAID HIKMAT ABD ALHAK

ESTACIONAMENTO INTELIGENTE

Orientadora: Professora M.C. Maria Marony Sousa Farias

Brasília

Junho, 2011

SAID HIKAMT ABD ALHAK

ESTACIONAMENTO INTELIGENTE

Trabalho apresentado ao Centro Universitário de Brasília como pré-requisito para a obtenção de Certificado de Conclusão do Curso de Engenharia de Computação.

Orientadora: Professora M.C.
Maria Marony Sousa Farias

Brasília

Junho, 2011

SAID HIKMAT ABD ALHAK

ESTACIONAMENTO INTELIGENTE

Trabalho apresentado ao Centro Universitário de Brasília como pré-requisito para a obtenção de Certificado de Conclusão do Curso de Engenharia de Computação.

Orientadora: Professora M.C. Maria Marony Sousa Farias

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 Fernandez
Coordenador do Curso

Banca Examinadora:

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

Prof. Msc Francisco Javier
UniCEUB

Prof. João Marcos Souza Costa
UniCEUB

*À meus pais, por não me deixar desistir
nunca, por maiores que sejam as
dificuldades. E por acreditar sempre que
iria vencer! E chegar hoje, até aqui.*

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus pela oportunidade de concluir mais uma etapa na minha vida.

Agradeço aos meus pais, Hikmat e Chárfite pelo apoio, confiança, amor e por sempre acreditarem nos meus sonhos e nas minhas vitórias.

Agradeço aos meus irmãos, Kamilo e Fádi, pelo companheirismo, amizade e pela ajuda nas horas difíceis.

Agradeço a minha namorada Kelly Amorim pelo seu amor, paciência e por ter me deixado conhecer esse seu lado vitorioso.

Agradeço aos meus amigos pela ajuda que me deram durante essa jornada, em especial Jose Carlos Cruz, Luciana Melo, Maria Luiza Braga e aqueles que de alguma forma me apoiaram direta e indiretamente para a concretização desse sonho.

Agradeço aos meus amigos de faculdade que me acompanharam durante essa trajetória.

Agradeço a minha Professora Orientadora Maria Marony Sousa Farias pela paciência e ajuda .

SUMÁRIO

LISTA DE FIGURAS.....	viii
LISTA DE QUADROS.....	x
LISTA DE ABREVIATURAS E SIGLAS.....	xi
RESUMO	xii
ABSTRACT	xiii
CAPÍTULO 1 – INTRODUÇÃO.....	14
1.1– <i>Motivação e Posicionamento.....</i>	15
1.2 – <i>Visão Geral do Projeto.....</i>	15
1.3 – <i>Objetivos.....</i>	16
1.4 – <i>Definição do Problema.....</i>	17
1.5 – <i>Escopo do projeto.....</i>	18
1.6 – <i>Diagrama de Blocos.....</i>	18
1.7 – <i>Metodologias</i>	19
1.8 – <i>Estrutura da Monografia</i>	19
CAPÍTULO 2 – FUNDAMENTOS TEÓRICOS.....	21
2.1 <i>Teoria do Smart Park.....</i>	23
2.1.1 – <i>Automação Predial (Smart Buildings).....</i>	24
2.2 – <i>Principais Problemas dos Estacionamentos Públicos</i>	24
2.3 – <i>Estacionamentos Automatizados Existentes no Mercado</i>	26
CAPÍTULO 3 – DESCRIÇÃO DE HARDWARE	29
3.1 – <i>Kit de desenvolvimento.....</i>	29
3.1.1 – <i>O Microcontrolador.....</i>	30
3.1.1.1 – <i>Regulador de Tensão 7805</i>	31

3.1.1.2 – Especificações do PIC 16F876A	32
3.1.1.3 – Pinagem do PIC 16F876A	33
3.1.1.4 - Kit de Gravação PICkit2®.....	34
3.1.2 – <i>Display</i> 7 segmentos	35
3.1.2.1 – Pinagem do <i>Display</i> de 7 Segmentos.....	39
3.1.3 – Comunicação Serial	40
3.1.3.1 – Pinagem do Conector Serial.....	41
Quadro 3.1 - Conector DB9 com designação de pinos.....	41
3.1.3.1– Conversor de Nível Serial para TTL	41
3.4 – <i>Infravermelho</i>	43
3.4.1 – Tipos de detecção e Aplicações.....	44
3.5 – <i>Circuito Infravermelho</i>	46
3.6 - <i>Circuitos de Inversão para os LEDs</i>	47
CAPÍTULO 4- TESTES E IMPLEMENTAÇÃO	49
4.1 – <i>Desenvolvimento do Projeto</i>	49
4.2 – <i>Topologia do Projeto</i>	50
4.3 – <i>Descrição da Implementação</i>	52
4.3.1 – Testes e Montagem da Maquete.....	52
4.3.2 – Programação de PICs em C	60
4.4 – <i>Apresentação do funcionamento da maquete</i>	61
CAPÍTULO 5 – CONCLUSÃO	63
5.1 – <i>Sugestões de Trabalhos Futuros</i>	64
REFERÊNCIAS BIBLIOGRÁFIA	65
APÊNDICE A	70

LISTA DE FIGURAS

Figura 1.1 – Diagrama de blocos do projeto.....	19
Figura 2.1 - Estacionamento vertical	23
Figura 2.2 – Estatística DETRAN.....	25
Figura 2.3 – Estacionamento automatizado do Morumbi Shopping.....	27
Figura 2.4 – Exemplo de estacionamento dos Estados Unidos.....	28
Figura 3.1 – Kit de desenvolvimento ACEPIC16F876A	29
Figura 3.2 – Fonte.....	30
Figura 3.3 – Kit de Desenvolvimento ACEPIC	31
Figura 3.4 – Regulador de tensão 7805	32
Figura 3.5 – Pinagem do PIC 16F87A	33
Figura 3.6 – Conector ICSP do PICkit2®.	34
Figura 3.7 – Gravadora PICkit2®.....	35
Figura 3.8 – Kit do Gravador PICkit2®.....	35
Figura 3.9 – Pinagem do display de 7 segmentos.....	36
Figura 3.10 - Tipos de displays, anodo comum e catodo comum	37
Figura 3.11 – Display multiplexado com baixa frequência	38
Figura 3.12 – Display multiplexado com alta frequência	38
Figura 3.13 – Conversão binária	39
Figura 3.14 - Representação do envio simultâneo de dados	39
Figura 3.15 – Pinagem do display de 7 segmentos.....	40
Figura 3.16 - Conector DB9 com sua pinagem enumerada	41
Figura 3.17- MAX232.	42
Figura 3.18 – Pinagem do MAX232	42
Figura 3.19 - Conexão da porta serial com o microcontrolador.....	43
Figura 3.20 - Detecção por reflexão.....	44
Figura 3.21 – Aplicação de detecção por reflexão	45
Figura 3.22 - Detecção por interrupção de feixe.....	45
Figura 3.23 – Aplicação de detecção por interrupção de feixe	46
Figura 3.24 – Circuito infravermelho	46
Figura 3.25 – Circuito receptor infravermelho	47
Figura 3.26 – Circuito de inversão	48

Figura 4.1 – Etapas do desenvolvimento do projeto.....	50
Figura 4.2 – Topologia do projeto.....	51
Figura 4.3 – Teste do brilho dos LEDs	53
Figura 4.4 – Interruptor de simulação aberto.....	54
Figura 4.5 – Interruptor de simulação fechado.....	54
Figura 4.6 – Simulação do circuito inversor	54
Figura 4.7 – Simulação dos três <i>displays</i>	55
Figura 4.8 - Simulação da placa de circuitos no programa Proteus.....	55
Figura 4.9 – Simulação do circuito infravermelho.....	56
Figura 4.10 – Higienização da placa	56
Figura 4.11 – Colagem da transparência à placa.....	57
Figura 4.12 – Passagem da tinta da transparência para a placa.....	57
Figura 4.13 – Processo de corrosão da placa.....	58
Figura 4.14 – Circuito desenhado na placa.....	58
Figura 4.15 - Remoção da tinta restante.....	59
Figura 4.16 – Perfuração da placa	59
Figura 4.17 - Componentes sendo soldados	59
Figura 4.18 – Placa pronta	60
Figura 4.19 – Maquete montada.....	60
Figura 4.20 – Extensões dos arquivos gerados	61
Figura 4.21 – Maquete finalizada.....	62

LISTA DE QUADROS

Quadro 3.1 - Conector DB9 com designação de pinos.	41
---------------------------------------------------------	----

LISTA DE ABREVIATURAS E SIGLAS

CD – Compact Disc

CD – Carrier Detect

CTS – Clear to Send

DB9 – Data Bus 9

DSR – Data Ser Ready

DRS – request to Send

DTR – Data Terminal Ready

EEPROM – Erasable Electronically Programmable Read Only Memory

GND – Ground, terra, aterramento

IBI – Intelligent Buildings Institute

ICSP – In Circuit Serial Programmer

LED – Light Emitting Diode

MCU – Microcontrolador

Mhz – Mega ou 10⁶ hertz, unidade de frequência

MIPS – Milhões de Instruções por Segundo

RAM – Random Access Memory

RI – Ring Indicator

RISC – Reduce Instruction Set Computer

RXD – Receive Data

TTL – Transistor-Transistor Logic

TXD – Transmit Data

USART – Universal Synchronous Asynchronous Receiver Transmitter

RESUMO

Neste trabalho é apresentado um protótipo de estacionamento inteligente com o objetivo de facilitar a vida dos condutores de veículos. Assim, na rampa de entrada do estacionamento haverá um display informando a quantidade de vagas disponíveis, servindo para direcionar o fluxo de veículos no estacionamento. Em cada setor, no decorrer do estacionamento haverá um display avisando a disponibilidade. No canto superior de cada vaga haverá dois LEDs (Light Emission Diode) indicativos, um vermelho e um verde e ainda um receptor de infravermelho. No local da vaga, em seu centro, haverá um sensor infravermelho para detecção de vaga ocupada. Esta detecção, se dá através da interrupção do feixe de luz emitido pelo sensor. Com a vaga ocupada, há o acionamento do LED vermelho. Contrariamente, o sinal que a vaga esta disponível é indicada pelo LED verde. A tecnologia proposta é, acessível e com baixo custo de implementação o que facilitará a vida dos condutores de veículos evitando o desperdício de tempo e combustível.

Palavras Chave: Infravermelho, estacionamento inteligente, PIC16F876A.

ABSTRACT

This paper presents a prototype of intelligent parking in order to make life easier of the drivers when they are on a parking lot. In the parking lot entrance there will be a display indicating the number of vacancies, serving as guide for the drivers and facilitating the traffic flow in the parking lot. In each sector of the parking lot there will be a display showing the vacancies of that sector. In the top of each parking space there are two LEDs (Light Emission Diode), one red and one green and an infrared receiver. There is an infrared emitter embedded in the ground in the parking space, as soon as a car parks the receiver won 't receive the light emitted by the emitter so the parking space will be occupied flashing the red LED. If the infrared receives the light/signal, it means that the parking space is available, and the LED will flash green. From this we can see that this is simple, accessible and low-cost technology, which can be easily implemented to make the lives of automobile drivers easier, avoiding wastes of time and fuel.

Keywords: Infrared: intelligent parking, PIC16F876A.

CAPÍTULO 1 – INTRODUÇÃO

O advento dos processos eletrônicos, nos dias atuais, trouxe velocidade e dinamismo à vida das pessoas em diversas áreas econômicas principalmente na comunicação, seja por meio de mensagens via videoconferências, rádio, internet, satélite. Porém para acompanhar esse ritmo de vida, cada vez mais acelerado são necessários que deslocamentos até o trabalho, faculdade, banco, academia, se inicie muitas vezes horas antes. Além do trânsito caótico e engarrafamentos, existe o problema cada vez mais freqüente, de encontrar vagas em estacionamentos. Segundo dados do DETRAN-DF, em 2000 a frota no Distrito Federal era de 585.424 veículos. Até fevereiro de 2011, este número saltou para 1.249.928. A questão é: onde estacionar todo este milhão de veículos? Indo mais além, o projeto foca no problema do controle de veículos dentro de um estacionamento inteligente.

Mesmo estando dentro de estacionamento privado não é fácil muito menos rápido a procura por uma vaga disponível. Neste trabalho é apresentado um sistema de controle visando facilitar a localização de vagas disponíveis em estacionamentos. A indicação das vagas disponíveis mais próximas de sua localização, para o condutor do veículo é feita através de um display na entrada do estacionamento. No projeto, por se tratar de um projeto de cunho acadêmico, são utilizados LEDs vermelhos e verdes: o LED verde indica vaga desocupada.

O problema de estacionamento é uma questão mundial. Já existem empresas produzindo ideias e soluções para o problema. Em Budapeste, já existe um grande projeto para a construção de estacionamentos subterrâneos por toda a cidade que usam além das indicações inteligentes de vaga, o deslocamento do veículo para espaços em várias direções do estacionamento, sendo movimentado por programas de computador a fim de encontrar a melhor posição a ser armazenado até que o condutor do veículo retorne. (BUDAPEST PARKING, 2010).

Nos Estados Unidos, desde 2007 foi atestada a importância desse tipo de estacionamento para a diminuição das dificuldades encontradas no mundo moderno. Perante esta situação, o tema a ser estudado trata do desenvolvimento de um estacionamento automatizado controlado pelo PIC e pelos sensores infravermelhos.

No Brasil, o Shopping Morumbi, em São Paulo, é o pioneiro na utilização desse sistema, chamado por alguns de estacionamento inteligente. Possivelmente em um futuro próximo, os sistemas automatizados estarão presentes em vários prédios comerciais trazendo economia, conforto e agilidade para as pessoas. Pensando nisso o escopo do trabalho é o desenvolvimento de um sistema para indique a localização de vagas disponíveis em estacionamento.

Para tanto se fez necessário a utilização de pesquisa e programação do PIC, pesquisa de sensor, infravermelho, receptor, LEDs, display 7 seguimentos, fios, protoboard, linguagem de programação C. Para o desenvolvimento do projeto foi necessária a utilização do simulador Proteus e a programação será feita utilizando a linguagem C para facilitar o uso dos componentes durante a instalação dos mesmos.

1.1– Motivação e Posicionamento

A construção de um estacionamento contribui para a concretização de um projeto que tem em sua essência o objetivo de gerar facilidade de acesso para o público alvo.

A motivação para a realização deste projeto surgiu a partir da observação do grande número de motoristas que procuram vagas ao tentar estacionar em ambientes comerciais, estádios, *shows* e *shoppings*.

Com o intuito de amenizar este problema e aumentar a agilidade, este projeto propõe, por intermédio de um protótipo, a construção de um estacionamento automatizado. Desta forma, foi proposto um sistema eficiente e inteligente no controle das vagas no ambiente do estacionamento, evitando assim engarrafamentos e a procura desnecessária.

1.2 – Visão Geral do Projeto

No protótipo desenvolvido, é simulado o controle de um estacionamento automatizado. Este protótipo estabelece que o controle das vagas sinalize através de um *display* de 7 segmentos a presença ou ausência de carros, alertando o motorista. Essa solução em forma de protótipo pode ser adequada à realidade com

o planejamento prévio em construções que queiram usufruir de estacionamentos automatizados.

Para uma melhor implementação deste projeto, o *display* servirá de painel que deverá estar localizado na entrada de cada setor, onde é possível visualizar a quantidade de vagas disponíveis no momento. Desta forma, evita-se congestionamento nos corredores, portanto, não será mais necessário procurar vagas, pois o motorista já saberá para qual setor poderá se direcionar devido à informação no *display*.

O sistema também deverá contar com um monitor na entrada principal mostrando a quantidade total de vagas livres no estacionamento, ou seja, haverá dois LEDs nas cores verde e vermelha, e um emissor de infravermelho e um receptor.

1.3 – Objetivos

O objetivo geral deste trabalho é apresentar um protótipo de um estacionamento automatizado utilizando um PIC. Esse projeto permite ao usuário saber de antemão onde estacionar logo ao entrar no estacionamento referido.

O software desenvolvido para o estacionamento indica a quantidade de vagas ocupadas ou disponíveis. Com base nos resultados referidos transmitirá os dados para um computador que mostrará o histórico das vagas e suas estatísticas, como o tempo de ocupação das vagas, juntamente com as vagas mais e menos ocupadas, e os horários de maior fluxo de carros.

Neste projeto, para a detecção de vaga ocupada ou desocupada, serão utilizados sensores de presença (transmissor e receptor infravermelho). Estas informações serão enviadas a um microcontrolador PIC. A quantidade de vagas disponíveis por setor será apresentada através dos *displays* de 7 segmentos, que tem como objetivo facilitar o fluxo de carros e diminuir o tempo gasto em filas.

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

- Construir um protótipo simulando um estacionamento;
- Criar um programa em C para controlar todo o sistema de vagas;

- Utilizar um PIC 16F876A para gerenciar a parte lógica do projeto;
- Utilizar sensores infravermelhos para informar a quantidade de vagas disponíveis;
- Utilizar LEDs para a indicação do *status* da vaga, na cor vermelha no modo ocupado e na cor verde no modo disponível;
- Elaborar um circuito que controle todas as vagas e os LEDs do estacionamento;
- Indicar através do *display* a quantidade de vagas disponíveis em cada setor e a também a soma total das mesmas.

1.4 – Definição do Problema

O problema estudado nesse projeto de pesquisa tem como objetivo apresentar uma solução viável para os transtornos causados pela procura de vagas em grandes estacionamentos.

As dificuldades encontradas para a ocupação das vagas em estacionamentos não vem sendo tratadas com a seriedade que o caso requer. Aqui em Brasília, por exemplo, em determinados locais, é praticamente impossível achar um espaço para estacionar o carro, até mesmo em estacionamentos privados, como em shoppings.

As soluções existentes para o caso em questão é o aumento de estacionamento e vagas correspondentes, bem como a utilização do estacionamento inteligente, assunto tratado no presente trabalho, para que o motorista não perca tanto tempo na procura das vagas.

Assim, o impacto negativo não é tão somente do motorista, vez que o meio ambiente também sofre com as consequências negativas dos problemas enfrentados no mundo moderno, sendo afetado cada vez mais por poluentes, degradando ainda mais o ar que respiramos.

No tocante aos benefícios da solução desse impasse destaca-se o melhor emprego do tempo do motorista, sempre atarefado com o cotidiano, bem como a melhoria do meio ambiente que recebe os gases poluentes decorrentes dessa perda de tempo procurando vagas, sem contar com a camada de ozônio que receberia menor quantidade de poluidores.

1.5 – Escopo do projeto

O escopo do projeto não contempla a automatização da entrada e nem da cobrança do tempo de ocupação da vaga no estacionamento.

A principal vantagem deste projeto é facilitar os motoristas a encontrarem vagas rapidamente em estacionamentos públicos.

O protótipo desenvolvido para a simulação do estacionamento não tem como solução de segurança a identificação do veículo estacionado, somente da vaga.

Outra questão de segurança que não foi abordada é a possibilidade de intrusão de pessoa desconhecida ou não autorizada no estacionamento.

1.6 – Diagrama de Blocos

O diagrama de blocos representa de forma objetiva a composição geral do projeto.

A figura 1.1 ilustra o diagrama de blocos deste projeto onde se pode observar dois parâmetros divididos em um quadro chamado vagas e em outro chamado controle.

O tópico vagas foi separado no quadro intitulado de detector que demonstra a atuação do sensor infravermelho quando há a presença e a ausência de carros.

Logo, quando o receptor não recebe o sinal do sensor significa que há um veículo estacionado, e assim o sinalizador vermelho é acesso, indicando que a vaga está ocupada. Porém, quando o receptor recebe o sinal do sensor infravermelho significa que a vaga está disponível e não há carro estacionado. Pois, neste caso o receptor do sensor está recebendo o sinal emitido pelo infravermelho ininterruptamente, ou seja, sem obstáculos, assim o LED verde permanecerá acesso até que um veículo interrompa o sinal.

O tópico controle foi dividido em dois quadros. O primeiro quadro realça a função do PIC que tem o controle das quantidades de vagas disponíveis no momento, e também tem a responsabilidade de enviar o histórico das mudanças do status das vagas. Portanto, o PIC tem o objetivo de controlar os painéis e mandar as informações para o PC escolhido. O segundo quadro destaca a importância do sistema do computador que armazena todas as informações de mudança de estado

das vagas e o registra os horários de ocupação e desocupação das mesmas. Então, o computador se ocupará de tratar os dados recebidos pelo PIC, tornando legíveis os dados para facilitar a análise do relatório, quando requerido pelo administrador do estacionamento.

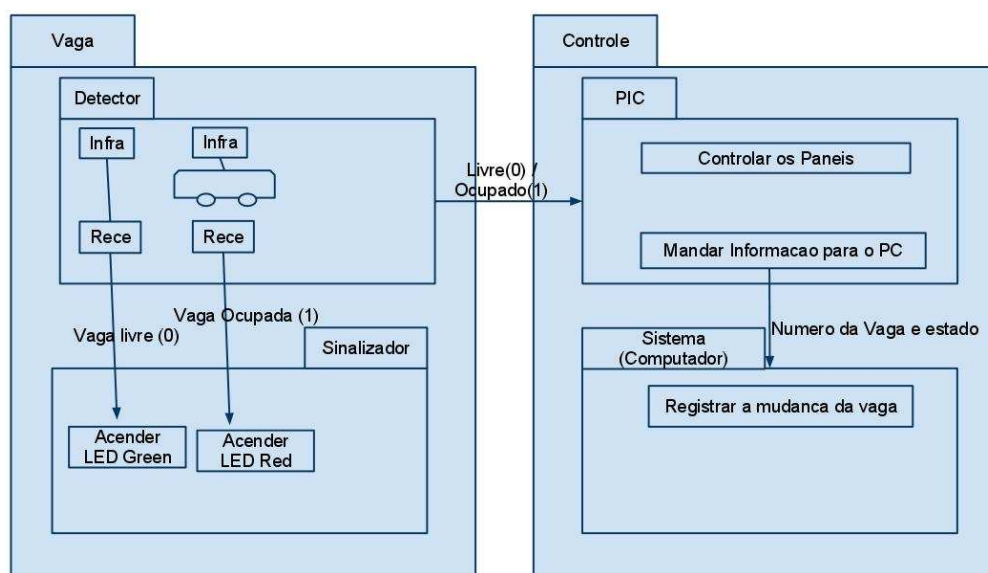


Figura 1.1 – Diagrama de blocos do projeto.

1.7 – 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 Eletrônicos, Circuitos e Máquinas Elétricas, LTP 1, LTP 2, Microprocessador e Microcontroladores, Sistemas de comunicação e Instalações Elétricas.

1.8 – 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 fazendo menção aos conceitos de *Smart Buildings*, estacionamentos automatizados.

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

Em 1808, com a vinda da Família Real para o Brasil, o país avançou consideravelmente no âmbito urbano. Historiadores estimam que cerca de 10 a 15 mil pessoas vieram para o Rio de Janeiro, explica Laurentino Gomes em seu famoso livro 1808. A cidade em princípio não estava preparada para tamanha acomodação, assim a solução veio com a urbanização da cidade: ampliação de ruas, veículos mais velozes, como as carruagens ganharam espaço no Rio de Janeiro. “No final do século XIX e no início do XX, aqui circulavam diferentes tipos de viaturas de tração animal – seges, traquitanas, berlindas, carruagens, caleças e tálburis, entre outros.” (MUSEU HISTÓRICO NACIONAL, 2011).

As importações de automóveis para o Brasil começaram por volta dos anos 20, trazendo ao país algum desenvolvimento. Mas o crescimento era bloqueado por ser o país apenas um produtor de matéria-prima. Em 1957, aos 15 de novembro, foi às ruas o primeiro automóvel fabricado no Brasil: a perua DKW. (CARROS ANTIGOS, 2011). Décadas se passaram e as noções de tempo, espaço e velocidade com que a informação circula passaram a ser revolucionários. Muitos hoje não conseguem imaginar suas vidas sem um automóvel para se locomover, tornou-se um membro extra do corpo humano.

SCHOR (1999) trata da “revolução invisível” fundamentada por Le Goff (1992) que explica justamente as revoluções no sentido de que a vida, seu ritmo, as noções de tempo e de espaço mudam completa e constantemente. “Invisível pelo fato de que estas técnicas entram de uma maneira sutil e impositiva em nossos cotidianos, em um movimento no qual tudo agora parece indispensável.”

Para quem tem um carro, estar nele e conviver com as questões diárias que seu uso acarreta são revoluções invisíveis. Mas algo que não está previsto para o futuro das revoluções invisíveis é o auto-controle quando já é tarefa árdua para os engenheiros de trânsito dominar o crescimento da frota de veículo e além disso, controlar o estacionamento desses veículos.

É preciso pensar em soluções menos imediatistas, capazes de resolver o problema atual do crescimento urbano e dar a novos motoristas mais qualidade de vida e conforto. Brasília, por exemplo, é a cidade planejada, mas foi pensada em

1960 quando a frota de veículos nem se comparava a complexidade do trânsito de hoje. Segundo notícia publicada no Estadão de São Paulo em novembro de 2010, a frota de veículos na capital paulista deveria chegar ao número impressionante de 7 milhões de veículos em janeiro deste ano.

Melhorar o transporte público ajuda a solucionar o problema do congestionamento diário, mas não resolve. O Brasil está longe de conseguir rotas e bons veículos que consigam locomover o passageiro com excelência até o local de destino. Além do mais, se a economia aumenta, a renda dos trabalhadores está aumentando, o que leva o cidadão a buscar mais comodidade. E aí que acarreta real possibilidade do aumento da frota automotiva.

Muitas vezes, o deslocamento de um local de origem ao local de destino em um transporte público pode levar até três vezes mais o tempo que levaria se a locomoção fosse realizada em veículo próprio. Porém, o tempo que o condutor leva para encontrar uma vaga para estacionar é tão absurdo que se torna tão inviável quanto se tivesse utilizado o transporte coletivo.

Para comportar tantos veículos são necessários excelentes estacionamentos com sistemas que facilitem o acesso tornando-o mais eficiente e mais ágil. O primeiro problema ainda é a falta de bons estacionamentos. Em reportagem do Correio Brasiliense que trata das dificuldades dos brasilienses para estacionar, o Detran-DF informou “que não cabe ao órgão construir estacionamentos, mas fiscalizar os já existentes.” E acrescentaram ainda que deste modo, os veículos estacionados em locais proibidos, continuariam sendo multados.

A verdade é que o motorista multado, infelizmente depende de seu veículo para o deslocamento diário, e provavelmente continuará recebendo multas e estacionando em locais proibidos por falta de lugar onde estacionar. Uma das soluções seria construir estacionamentos que ocupem menos espaço, como arranha-céus que funcionam apenas como estacionamentos.

Mundo a fora, empresas buscam soluções para este problema constante de estacionamento e tornar o acesso a eles muito mais fácil e automatizado. Pensando nisso, já foram criados estacionamentos adaptados a essa nova realidade como é o caso dos estacionamentos verticais. Os estacionamentos verticais são um modelo de edifício-garagem em que o sistema de condução dos veículos é totalmente

automatizado. Os usuários não têm acesso ao interior, onde os carros são guardados, já que os veículos são direcionados para as vagas por meio de equipamentos totalmente robotizados. (METALICA, 2011). A figura 2.1 ilustra este tipo de estacionamento.



Figura 2.1 - Estacionamento vertical.
(Fonte: METALICA, 2011).

Esses estacionamentos de grande porte estão fora da realidade atual no Brasil. Conta-se então, com o advento de estacionamentos com sistemas que melhoram o deslocamento dentro dos mesmos. Pensando nesses novos estacionamentos capazes de suportar elevada quantidade de veículos e, que possuem alta rotatividade de seus usuários este trabalho apresenta um sistema de localização de vagas disponíveis com direção e quantidade.

2.1 Teoria do Smart Park

O trabalho traz um estudo sobre a domótica e suas principais funções. Com o propósito de melhorar a qualidade de vida, reduzir o trabalho doméstico, aumentar o bem-estar e a segurança.

2.1.1 – Automação Predial (Smart Buildings)

O conceito de *Smart Building* ou “Edifício Inteligente” apareceu nos EUA por volta da década de 80. De acordo com Camargo & Neves (2002, p. 2), um Edifício Inteligente é “aquele que utiliza a tecnologia para diminuir os custos operacionais, eliminar os desperdícios e criar uma infra-estrutura adequada para aumentar a produtividade dos usuários”.

Nos Estados Unidos a *Intelligent Buildings Institute* (IBI) define *Smart Building* como: “aquele que oferece um ambiente produtivo e econômico através da otimização de quatro elementos básicos: estrutura, sistemas, serviços e gerenciamento; bem como das inter-relações entre eles”. Dessa forma, um Edifício Inteligente promove a transferência de dados de um sistema para outro.

De acordo com Camargo & Neves (2002, p. 3), o conceito de Edifício Inteligente começa a ser utilizado no Brasil a partir de 1986 com a inauguração do Edifício Citicorp/Citibank, localizado em São Paulo. Tratava-se do primeiro grande edifício de escritórios com destacado nível de monitoramento e controle energético.

2.2 – Principais Problemas dos Estacionamentos Públicos

Um dos problemas da modernidade e da facilidade e acesso aos automóveis sem dúvida têm como principal consequência a falta de lugares para estacionar. Isso envolve a política pública de trânsito, ou seja, se torna um problema tanto social quanto governamental.

A quantidade de vagas existentes não comporta o número de carros na rua. No Distrito Federal, por exemplo, segundo dados do Detran, em 10 anos, de 2000 a 2010 a frota duplicou e até fevereiro de 2011 contava com mais de um milhão e duzentos mil carros. A Figura 2.2 ilustra a realidade no aumento da frota automotiva no Distrito Federal.

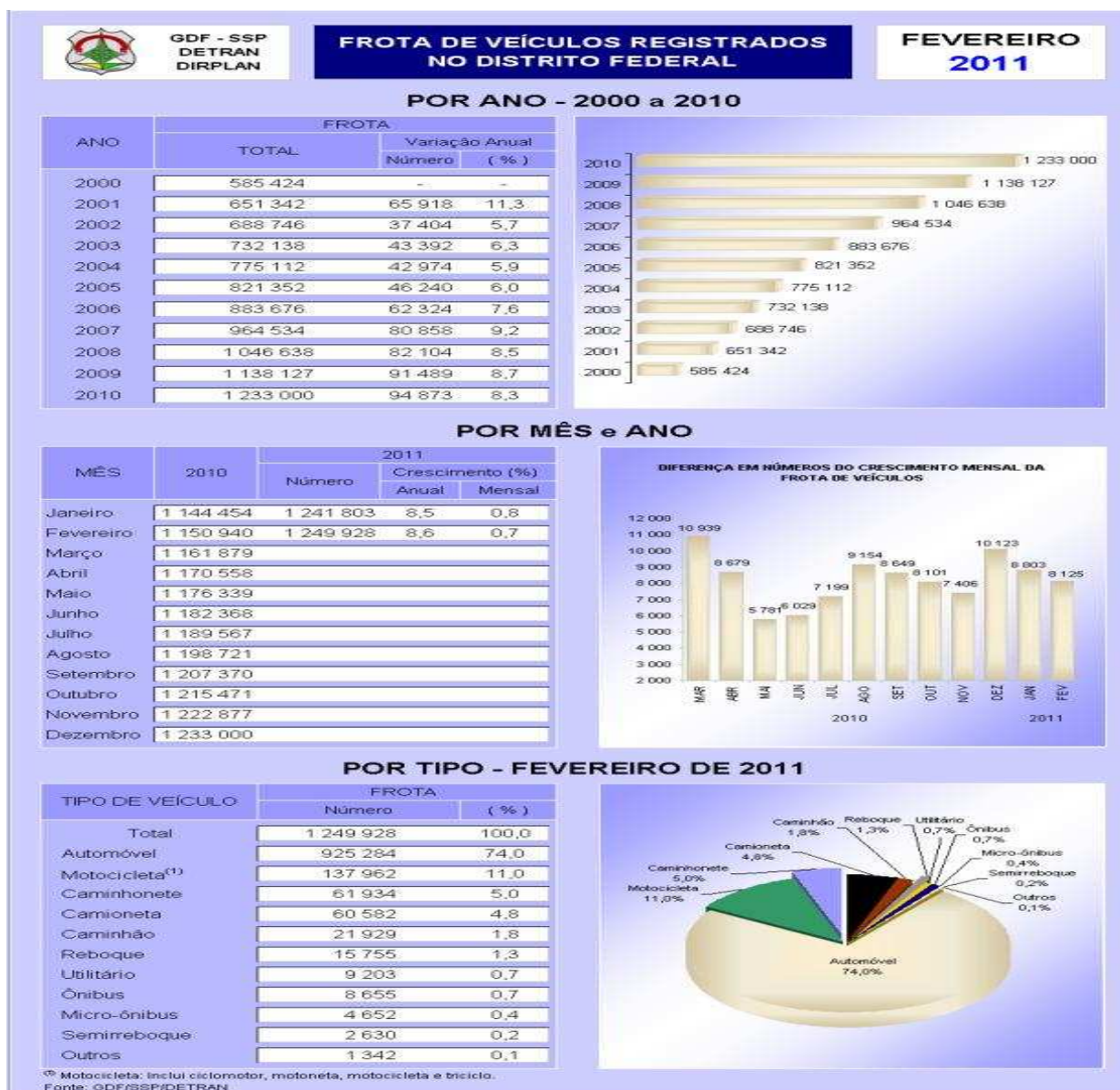


Figura 2.2 – Estatística DETRAN.
(FONTE: DETRAN, 2011).

O fato é que quanto mais a frota aumenta crescem os problemas com estacionamento, seja na seara pública ou privada. Infelizmente são poucos os estacionamentos públicos verificados, em especial, do Distrito Federal e quando existem são lotados dia e noite, sem acesso para estacionar.

No caso de Brasília, existe muito fluxo de pessoas trabalhando nos Centros Comerciais Sul e Norte. Nestes locais estão concentrados alguns estacionamentos públicos, contudo, dificilmente se acha uma vaga. Cabe ressaltar ainda que o

governo muitas vezes fica engessado, vez que não pode mudar a estrutura da cidade por ser tombada pelo patrimônio histórico.

Em reportagem no correio brasileiro do dia 13 de fevereiro de 2011, o “Professor de engenharia de tráfego na Universidade de Brasília (UnB), Paulo César Marques explica que o crescimento da frota em função da melhora do poder aquisitivo das pessoas é uma coisa a ser comemorada. “O que é nocivo é a dependência do veículo e o uso que se faz dele. Do jeito que está, não haverá espaço possível para a circulação e ampliação de estacionamentos que dê jeito”, atesta. Marques lembra que alguns países da Europa têm taxas de motorização até maiores que a de Brasília. Mas as pessoas usam o metrô, o ônibus e a bicicleta. “Aqui, temos uma política muito permissiva ao uso do transporte individual” (CORREIO BRAZILIENSE, 2001).

O engenheiro de tráfego que foi entrevistado pelo jornal indicado lembra que em pouco tempo não restará mais espaço para os carros existentes na cidade e que em outras cidades do mundo existe taxa de motorização, ou seja, mais veículos para cada habitante, e que existe uma política de transporte público mais eficiente que o nosso, acarretando a menor quantidade de carros trafegando nas ruas.

Estacionamentos públicos, em sua grande maioria, não têm espaço satisfatório que abarque as vagas necessárias, não tem fiscalização e nem mesmo segurança. Contudo, o foco da presente monografia está voltado para a falta de vagas, de espaço para os motoristas estacionarem seus veículos e não ficarem dando voltas a procura, tendo como consequência o trânsito congestionado e o gasto de combustíveis, piorando o meio ambiente com os gases emitidos.

2.3 – Estacionamentos Automatizados Existentes no Mercado

O primeiro shopping do Brasil a adotar um sistema de estacionamento automatizado foi o Morumbi Shopping localizado em São Paulo capital, conforme mostrado na Figura 2.3.



Figura 2.3 – Estacionamento automatizado do Morumbi Shopping.

(FONTE: PRADA, 2010).

Este estacionamento utiliza sensores e LEDs verde e vermelho, que buscam acabar com o tempo perdido em busca de vagas para estacionar no shopping. Da mesma forma, o sistema ajuda a administração do shopping a fazer um mapa de ocupação e o controle do fluxo de veículos.

Nesse estacionamento, a visualização dos painéis indicativos para a localização de vagas pode ser feita a uma grande distância, logo na entrada e também em demais setores. Esses painéis fazem a contagem de vagas disponíveis em cada setor e mostram ao motorista onde encontrá-las.

A distribuição de sensores se encontra em cada vaga individualmente, quando este detecta a presença de um veículo a luz vermelha do *dommi* é acessa indicando que o local está ocupado. A luz vermelha só se apagará para que a luz

verde se acenda quando o local da vaga se tornar vago para um carro, como mostrado na figura 2.3.

De acordo com (PRADA, 2010), no estacionamento do Morumbi Shopping as vagas especiais, para idosos ou portadores de necessidades, tem um diferencial das outras. Estas possuem o LED indicativo da cor azul, e contam com alto-falantes que alertam o motorista que a vaga está reservada para um cliente preferencial, evitando possíveis transtornos.

A figura 2.4 é um exemplo de estacionamento nos Estados Unidos utilizando esta tecnologia.



Figura 2.4 – Exemplo de estacionamento dos Estados Unidos.

(FONTE: SOMNUSCARS, 2009).

CAPÍTULO 3 – DESCRIÇÃO DE HARDWARE

Este capítulo apresenta às especificações dos dispositivos utilizados e são explicados os circuitos necessários para a montagem do protótipo deste projeto. Para o correto desenvolvimento do objetivo deste trabalho, que é apresentar um protótipo de um estacionamento automatizado, foram utilizados os seguintes hardwares: o Kit de desenvolvimento contendo o microcontrolador, o regulador de tensão, *display* de 7 segmentos e o conector serial DB9. Além disso, foram utilizados também, circuitos de infravermelho e circuitos de inversão para os LEDs (*Light Emitting Diode*).

3.1 – Kit de desenvolvimento

O kit de desenvolvimento adquirido para o projeto foi o kit de desenvolvimento ACEPIC16F876A. Nele são acoplados vários dispositivos dentre os quais foram utilizados: o microcontrolador, o regulador de tensão, *display* de 7 segmentos e o conector serial DB9. Na figura 3.1 é mostrada a imagem desse kit utilizado na maquete.

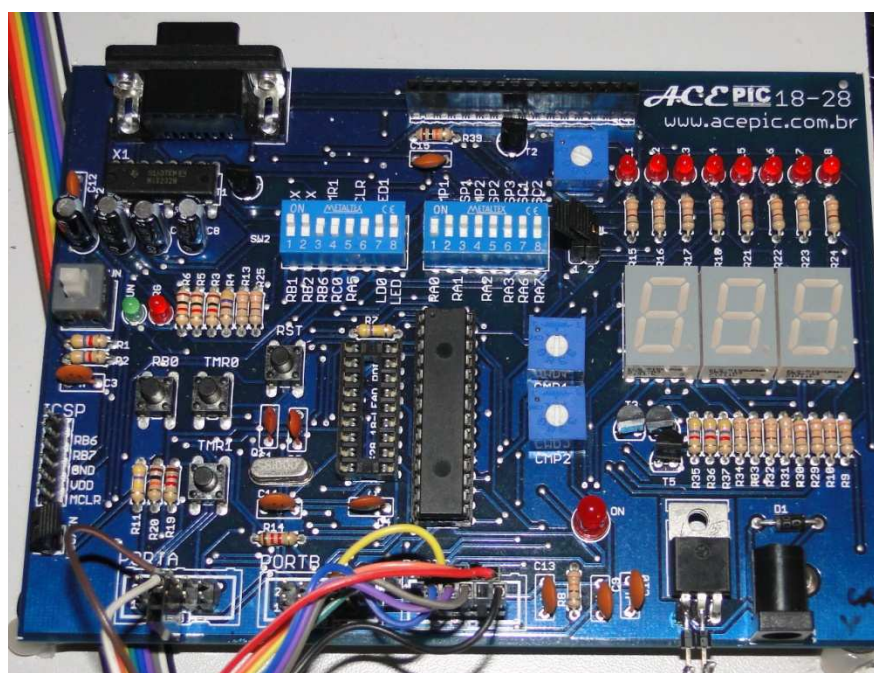


Figura 3.1 – Kit de desenvolvimento ACEPIC16F876A.

3.1.1 – O Microcontrolador

Microcontrolador é um computador em um único chip otimizado para controlar dispositivos externos. Suas vantagens são a auto-suficiência e baixo custo (JOSÉ, A. 2009). E foi este dispositivo o escolhido para controlar os demais hardwares do projeto. Para tanto, foi adquirido o kit de desenvolvimento ACEPIC16F876A que contém o microcontrolador 16F876A. Este microcontrolador é responsável pelo controle dos infravermelhos, dos três *displays* de 7 segmentos e da comunicação serial.

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 400m 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 3.2 ilustra a imagem dessa fonte usada no projeto.



Figura 3.2 – Fonte.

Já a figura 3.3 é o desenho do Kit de desenvolvimento citado acima e suas ramificações.

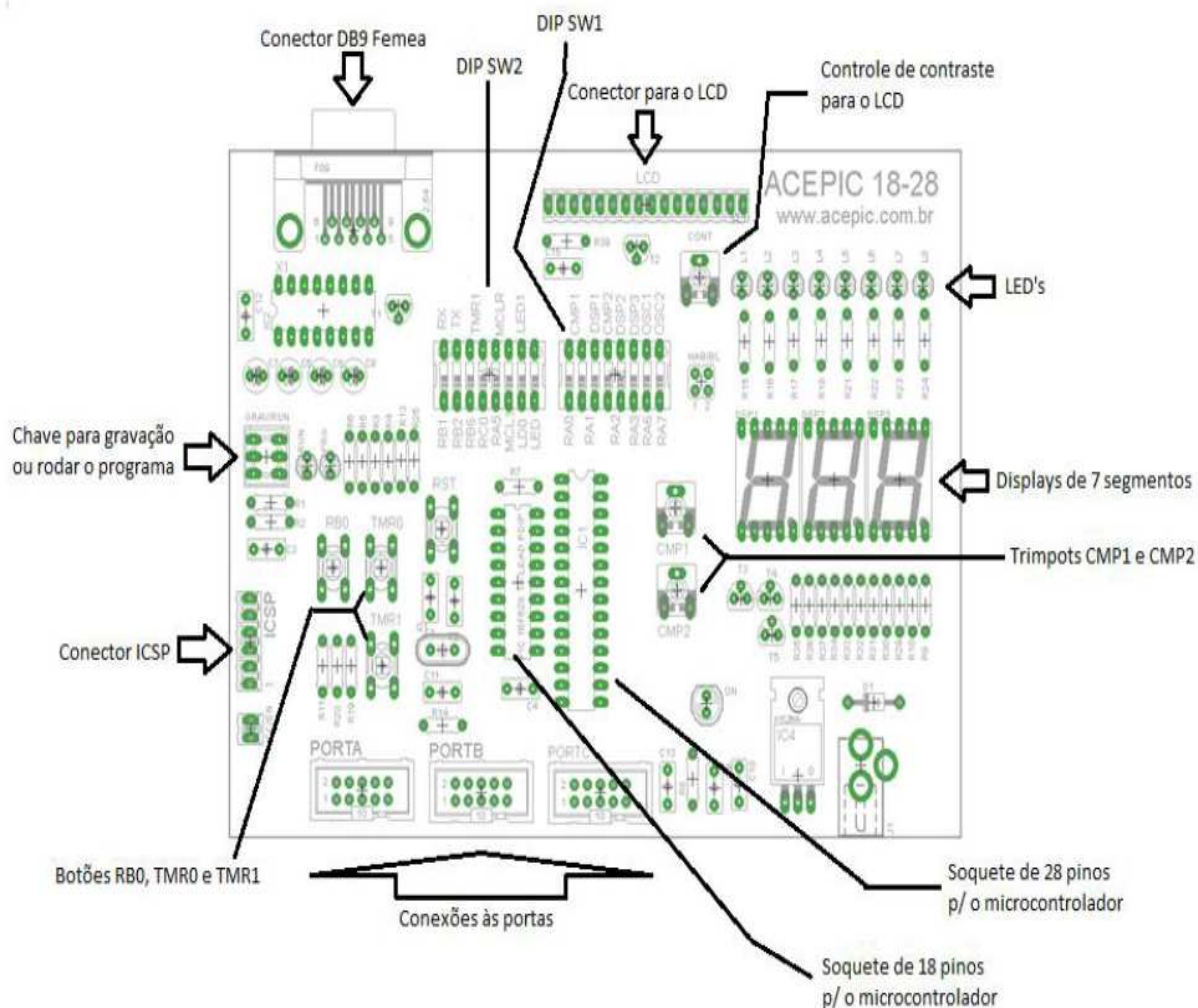


Figura 3.3 – Kit de Desenvolvimento ACEPIC.

(FONTE: ACEPIC, 2011).

3.1.1.1 – Regulador de Tensão 7805

O regulador de tensão utilizado foi o de numero 7805 da série 78XX. Ele regula a tensão do Kit de desenvolvimento, de 12 V para 5V, alimentando todo o sistema. Este regulador contem 3 pinos: o de entrada, o de saída e o aterramento. A figura 3.4 ilustra o regulador 7805 e seus pinos.

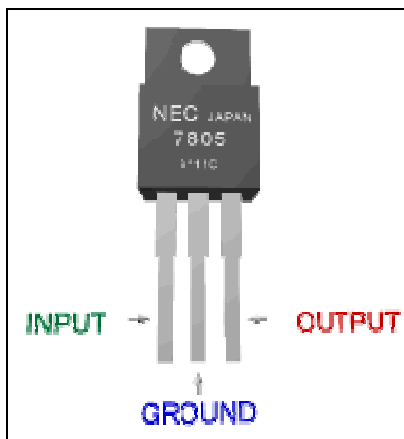


Figura 3.4 – Regulador de tensão 7805.

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

Pino 2: GROUND – Terra;

Pino 3: OUTPUT - Saída regulada de 5V.

3.1.1.2 – Especificações do PIC 16F876A

O PIC16F876A fabricado com a tecnologia CMOS foi escolhido para o projeto devido ao baixo custo e por conter a pinagem necessária para implementação em questão. Seguem algumas especificações deste dispositivo:

Processador RISC (Reduce Instruction Set Computer)

→ 35 instruções de 14 bits;

→ Frequência máxima de funcionamento - 20Mhz (frequência do cristal);

→ Cada ciclo de relógio corresponde à frequência do cristal / 4 = 5us, efetuando a cada segundo 5 MIPS (milhões de instruções por segundo);

→ Tempo de execução das instruções: 1 ciclo de relógio.

Características da memória;

→ Memória de programa (FLASH) de 8K (word) de 14 bits;

→ Cada instrução é codificada numa word de 14 bits;

→ Memória de dados RAM de 368 bytes;

→ Memória de dados EEPROM de 256 bytes.

Características de periféricos:

- 22 linhas de entrada/saída, agrupadas em 3 ports (PORTA, PORTB e PORTC);
- Conversor analógico à digital de 10 bits, com um máximo de 5 canais de entrada analógica;
- USART (Universal Synchronous Asynchronous Receiver Transmitter);
- 13 tipos de interrupções, por exemplo externa RB0/INT, TMR0.

3.1.1.3 – Pinagem do PIC 16F876A

A Figura 3.5 ilustra a pinagem do PIC 16F87A e em seguida são especificadas as ligações destes pinos.

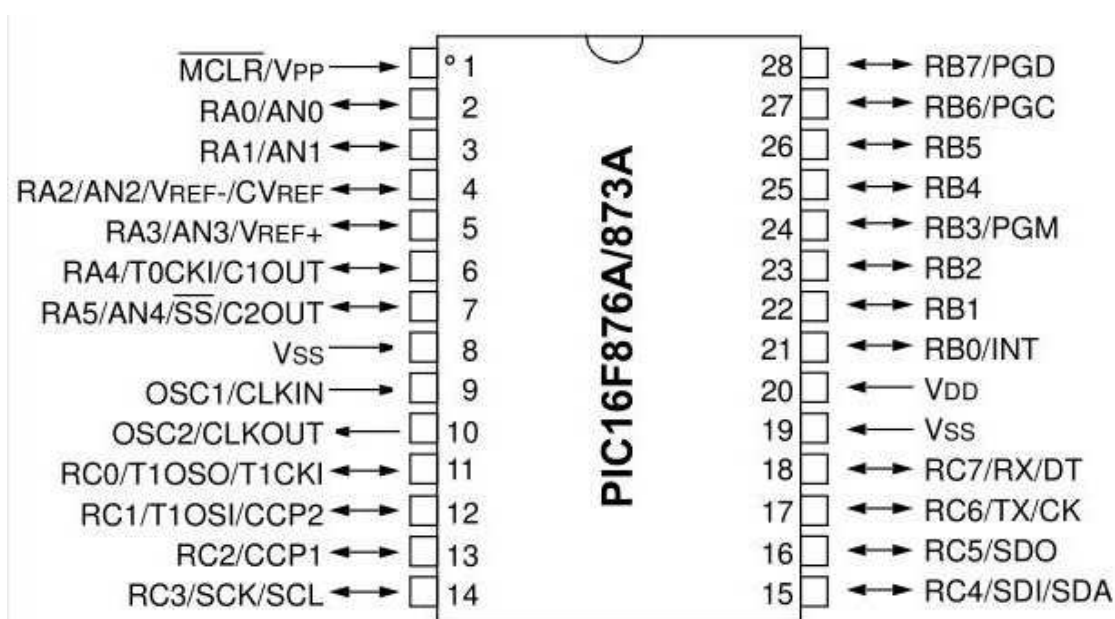


Figura 3.5 – Pinagem do PIC 16F87A.

- Pino 1 – ligado a pino de programação (*reset*);
- Pino 3 – ligado ao pino 14 do *display* 1;
- Pino 4 – ligado ao pino 14 do *display* 2;
- Pino 5 – ligado ao pino 14 do *display* 3;
- Pinos 2, 9, 10 e 18 – não estão sendo utilizados;
- Pino 6, 7 – ligados ao Port A do kit de desenvolvimento;
- Pinos 11,12,13,14,15,16,18 – ligados ao Port C do kit de desenvolvimento;

- Pino 17 – é o pino que transmite o log para a porta serial;
- Pinos 21 a 27 – ligados ao Port B do kit de desenvolvimento.

3.1.1.4 - Kit de Gravação PICkit2®

Composto pelo hardware de gravação ICSP (*In Circuit Serial Programmer*) para o PIC e uma interface de controle do gravador que depois de compilado no MPLAB o código fonte em C ou .asm, gera o arquivo .Hex, sendo este arquivo enviado à memória do microcontrolador PIC pelo PICkit, com a transferência do arquivo .Hex o PIC é gravado.

O programa PICkit2® da Microchip® na versão 2.61 atualiza e grava qualquer linguagem de programação. Além de contar com analisador lógico de três canais e conversor. Possui TTL sobre USB que permite comunicar o computador diretamente com PIC.

O programa é uma ferramenta de desenvolvimento com uma interface fácil de usar para programação e depuração em famílias do PIC. O conector ICSP, mostrado na figura 3.6, suporta PICs de vários pinos.

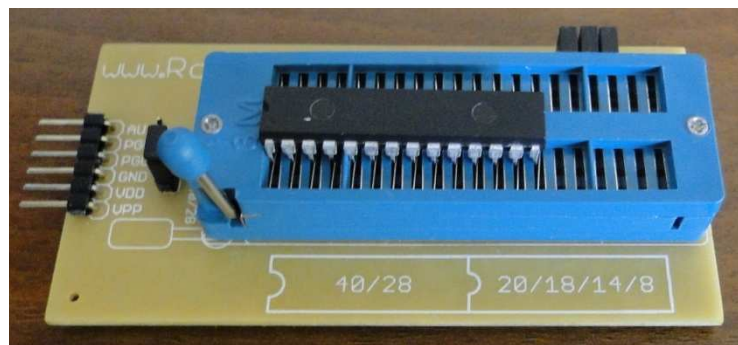


Figura 3.6 – Conector ICSP do PICkit2®.

A figura 3.7 ilustra o kit gravador PICkit2®.

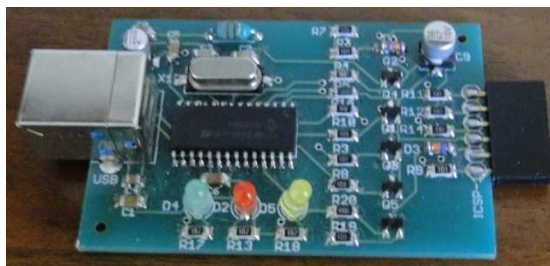


Figura 3.7 – Gravadora PICkit2®.

O software possui o objetivo de identificar o microcontrolador a ser gravado. Funciona em qualquer computador com uma porta USB disponível de 1.1 ou 2.0.

O kit adquirido, constatado na figura 3.8, além de possuir o Gravador PICkit2®, contém um cabo USB, um conector ICSP, e um CD com os softwares para Windows®, Linux® e MAC OS®, além das principais linguagens de programação e seus manuais.

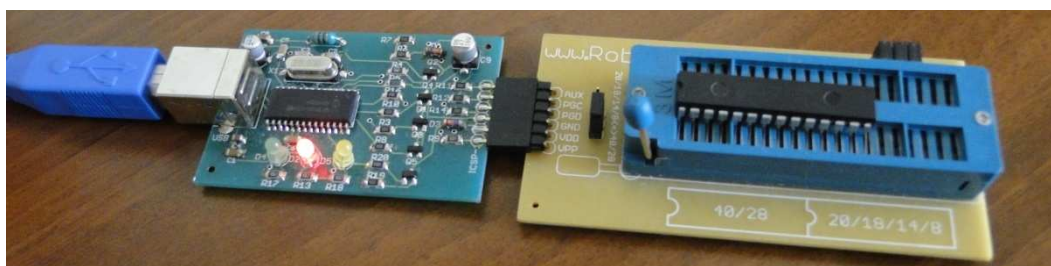


Figura 3.8 – Kit do Gravador PICkit2®.

3.1.2 – *Display* 7 segmentos

Os *displays* de 7 segmentos são comumente usados em eletrônica como forma de exibir uma informação numérica sobre as operações internas de um dispositivo.

Este *display* é composto de 7 elementos dispostos em forma de oito, os quais podem ser ligados ou desligados individualmente. Eles podem ser combinados para produzir representações simplificadas de algarismos arábicos.

Displays de 7 segmentos podem usar matrizes de LEDs e outras técnicas de controle ou geração de luz, tais como descarga de gás de catodos frios, vácuo fluorescente, filamentos incandescentes e outros.

O *display* pode ser do tipo anodo comum, ou seja, os terminais anodo de todos os segmentos estão interligados internamente e para o *display* funcionar, este terminal comum deverá ser ligado em Vcc, enquanto que o segmentos para ligar precisam estar ligados no GND (lógica inversa). Já o *display* catodo comum, é o contrário, ou seja, o terminal comum deverá ser ligado ao GND e para ligar os segmentos é necessário aplicar Vcc aos terminais. Atualmente, o *display* mais comercializado é o do tipo anodo comum (TOCCI, 2003). Apesar disso, o *display* de 7 segmentos utilizado no projeto foi o catodo comum.

Como os segmentos utilizados no projeto são leds, foi preciso limitar a corrente, para isso foi usado um resistor em cada segmento. A corrente utilizada, depende do brilho que se quer no *display*, no projeto foi utilizado 330 ohms, para uma fonte de 5Volts. O que equivale entre 15mA. Não deve ser usado valores de resistores muito baixo, pois estaria assim reduzindo a vida útil do *display*. A figura 3.9 apresenta um *display* de 7 segmentos e sua respectiva pinagem.

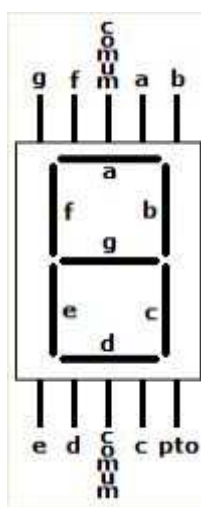


Figura 3.9 – Pinagem do display de 7 segmentos.

(FONTE: PUCRS, 2011).

Já a figura 3.10 ilustra os dois tipos de *displays*, anodo comum e catodo comum.

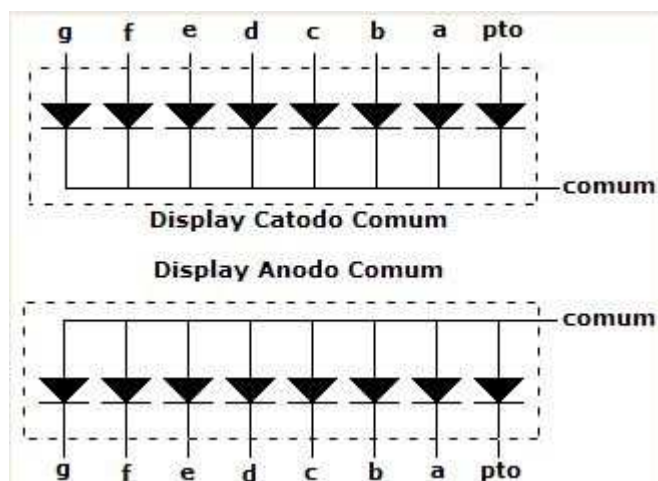


Figura 3.10 - Tipos de displays, anodo comum e catodo comum.

(FONTE: PUCRS, 2011).

O *display* acoplado no kit de desenvolvimento é um *display* multiplexado. Em um *display* multiplexado apenas um dígito fica ligado de cada vez em rápida sucessão. Se a multiplexação for bastante rápida, a persistência da visão cria a impressão de que todos os dígitos estão ligados simultaneamente. Um visor multiplexado tem as seguintes vantagens sobre um não multiplexado:

- Economia de hardware: É necessário apenas um registrador e um decodificador para 7 segmentos para todos os dígitos. No sistema não multiplexado seria necessário um registrador e um decodificador para cada dígito;
- Economia de energia O *display* de LED tem maior eficiência luminosa quando é usado em modo pulsado;
- Redução do número de linhas de interconexão. Um computador pode se comunicar com um visor ABCD multiplexado de 4 dígitos com apenas 6 fios: 4 para o código ABCD do dígito e mais 2 para selecionar o dígito. Para um visor de 8 dígitos seria necessário apenas um fio adicional.

E as desvantagens de multiplexar o *display* são a maior complexidade do circuito e a redução da luminosidade aparente. A Figura 3.11 ilustra um *display* multiplexado com baixa frequência. E a figura 3.12, um *display* multiplexado com alta frequência onde se pode observar que ao olho humano todos os dígitos parecem estar ligados simultaneamente (TOCCI, 2003).

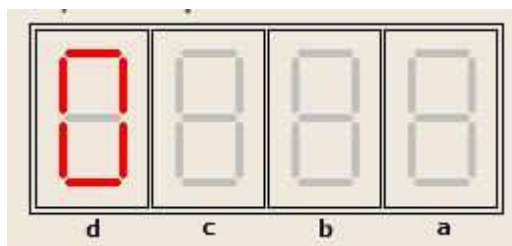


Figura 3.11 – Display multiplexado com baixa frequência.
(FONTE: PUCRS, 2011).

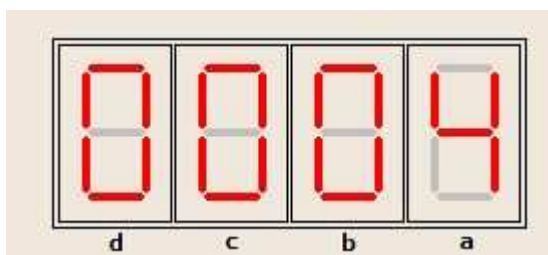


Figura 3.12 – Display multiplexado com alta frequência.
(FONTE: PUCRS, 2011).

A comunicação do *display* é feita em base hexadecimal, para isso foi necessária a construção de uma rotina que recebesse um dígito em hexadecimal e devolvesse o correspondente valor para ativação de cada um dos segmentos do *display*. Este valor deverá ser enviado para o Port B do kit de desenvolvimento. A figura 3.13 mostra a correspondência entre os dígitos em binário de entrada e o valor a enviar para cada um dos 7 segmentos do *display*. Por exemplo, para visualizar o dígito 7 é necessário que o segmento a,b e c, ilustrados na figura 3.13, estejam ativos, isto é, mover para o PORTB o valor em binário b'00001110'. E o mesmo ocorre com os outros dígitos mudando apenas os segmentos que serão ativados.

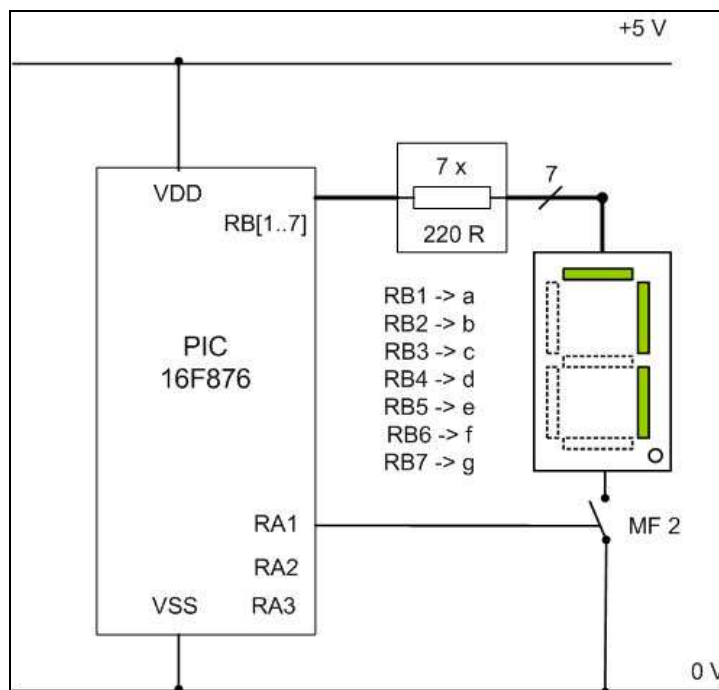


Figura 3.13 – Conversão binária.

(FONTE: SWEET, 2011).

Como ilustrado na figura 3.14 pode-se notar que todos os segmentos dos *displays* estão ligados à saída B (RB[1..7]) do kit de desenvolvimento. Os dados são mandados para os *displays* simultaneamente. Estes dados devem ser atualizados periodicamente com uma taxa fixa (taxa de refreshment aprox. = 10ms). A figura 3.14 mostra este envio simultâneo de dados aos *displays*.

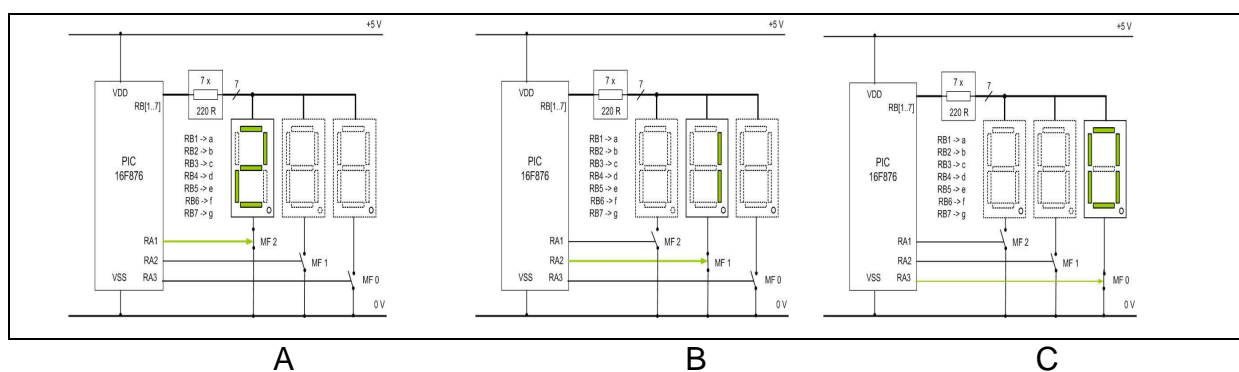


Figura 3.14 - Representação do envio simultâneo de dados.

(FONTE: PUCRS, 2011).

3.1.2.1 – Pinagem do *Display* de 7 Segmentos

A pinagem do *display* de 7 segmentos é feita como ilustra a Figura 3.15.

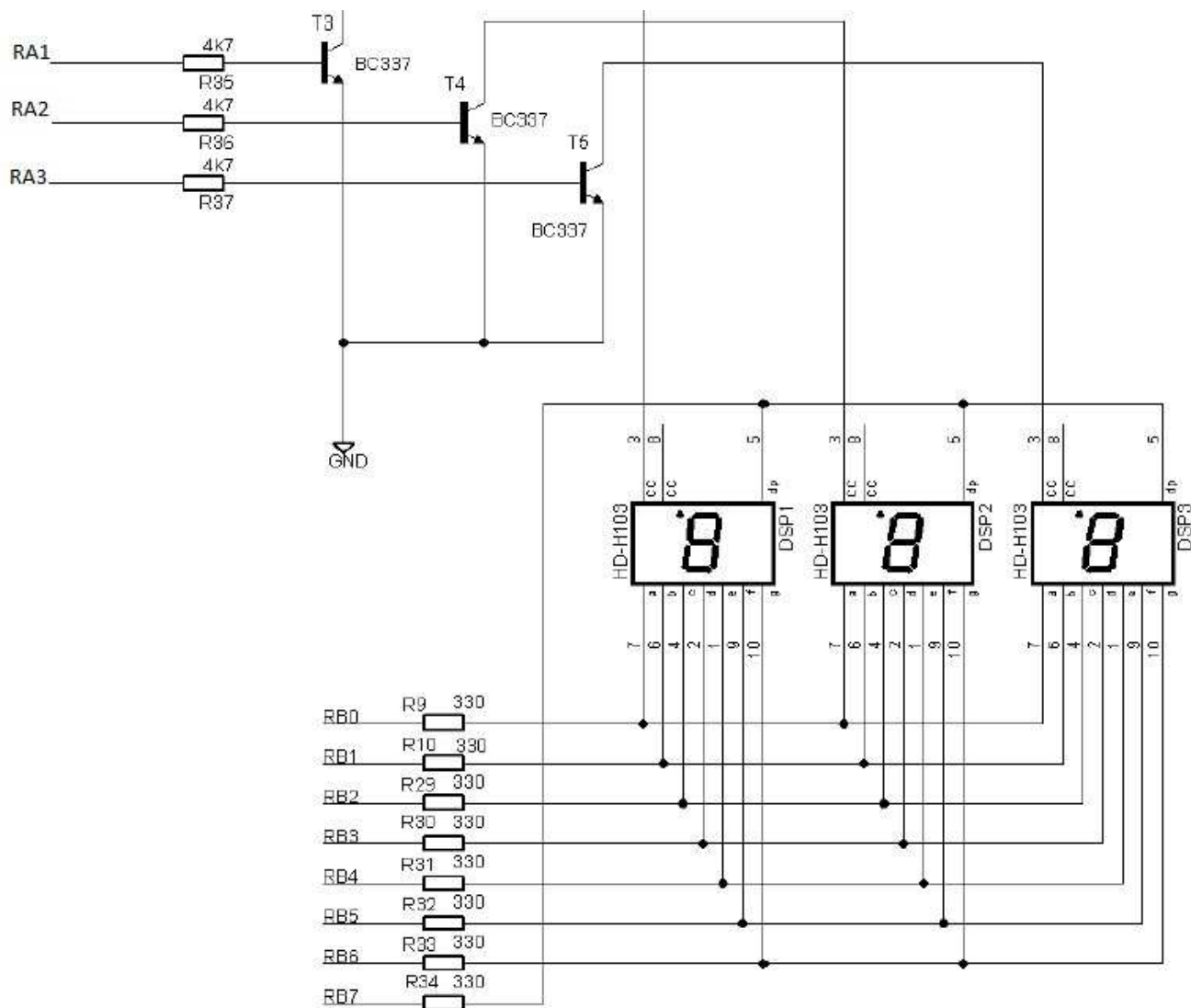


Figura 3.15 – Pinagem do display de 7 segmentos.

Os pinos 7,6,4,2,1,9,10 e 5 são ligados respectivamente nas portas do kit de desenvolvimento RB0, RB1, RB2, RB3, RB4, RB5, RB6 e RB7. Para estas ligações são necessários resistores de 330 Ω . Estes Ports está ligado ao PIC 16F876A.

Os pinos 3 de cada *display* estão ligados as portas do kit RA1, RA2 e RA3. Para estas ligações são necessárias resistores de 4,7K Ω .

3.1.3 – Comunicação Serial

A comunicação serial é o processo de enviar bits de cada vez, seqüencialmente, 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 3.16.

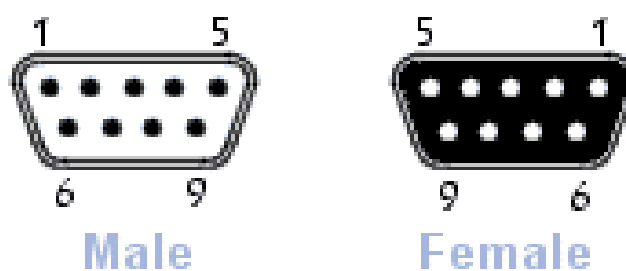


Figura 3.16 - Conector DB9 com sua pinagem enumerada.

(FONTE: KIOSKEA, 2011)

A porta serial permite a comunicação entre o kit de desenvolvimento e o software utilizado no projeto.

3.1.3.1 – 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 campanha

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).

3.1.3.1– 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 3.17, 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 3.17 é uma foto do MAX232 utilizado no projeto.

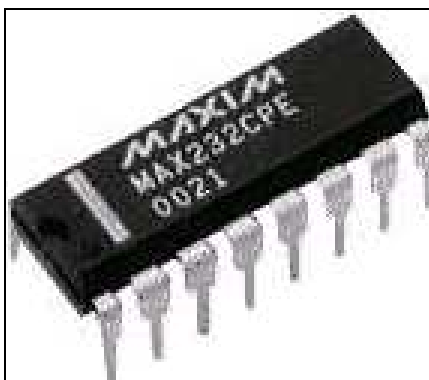


Figura 3.17- MAX232.
(FONTE: CIRCUIT SHOP, 2011)

A Figura 3.18 mostra a pinagem do MAX232.

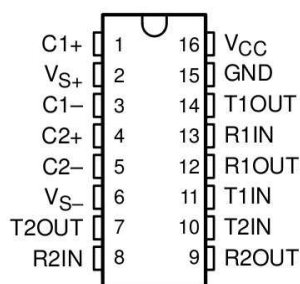


Figura 3.18 – Pinagem do MAX232.
(FONTE: TEXAS INSTRUMENTS, 2004)

Pino 1 e 3: capacitor de 1 μ F;

Pino 2 e 16: capacitor de 1 μ F;

Pino 4 e 5: capacitor de 1 μ F;

Pino 6: capacitor de 1 μ 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 μF para desclapamento.

A figura 3.19 ilustra a conexão da porta serial com o microcontrolador e a função do MAX232 nesta conexão.

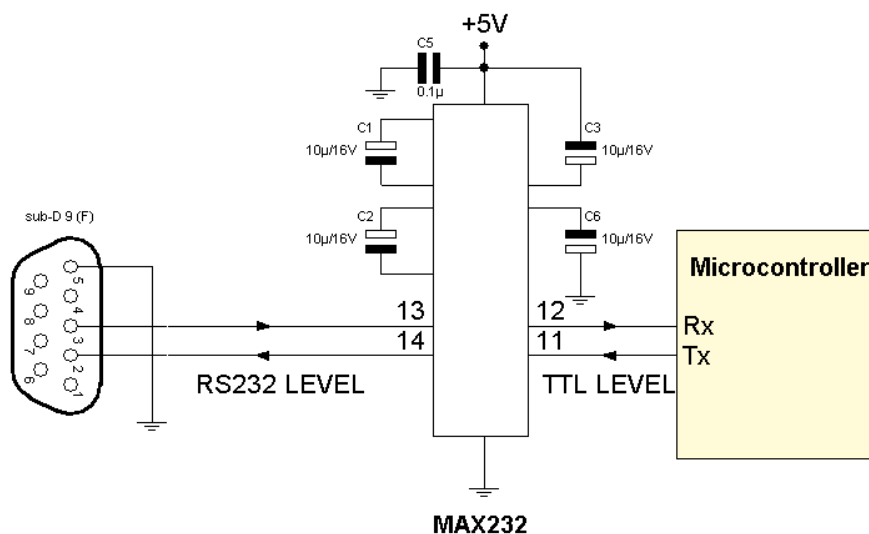


Figura 3.19 - Conexão da porta serial com o microcontrolador.

3.4 – Infravermelho

Por ser uma tecnologia de simples implementação os sinais de infravermelho foram escolhidos para detectar a presença de um carro em uma vaga na maquete montada. Esta tecnologia possui a vantagem de ser de baixo custo e de não necessitar de contato físico.

3.4.1 – 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 3.20 mostra um exemplo de detecção por reflexão (THOMAZINI, 2005; URBANO BRAGA, 2005).

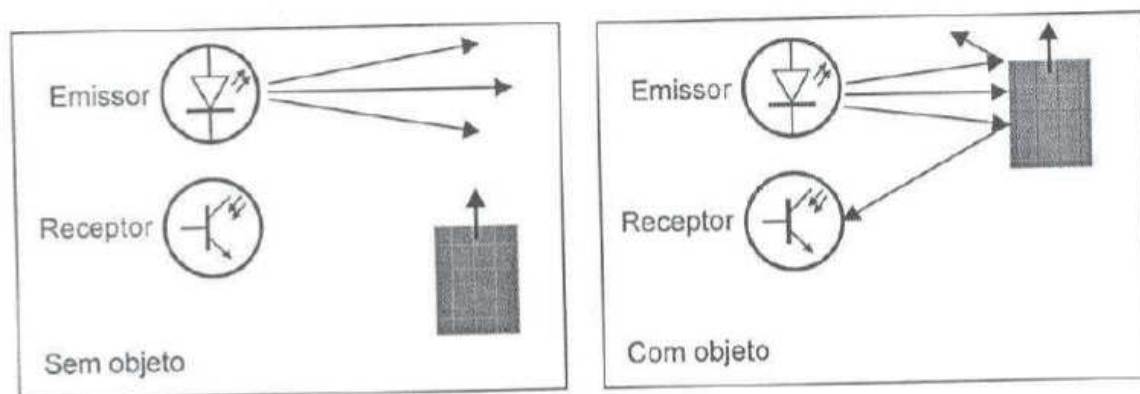


Figura 3.20 - Detecção por reflexão.

(FONTE: THOMAZINI, 2005; URBANO BRAGA, 2005).

A figura 3.21 mostra 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.

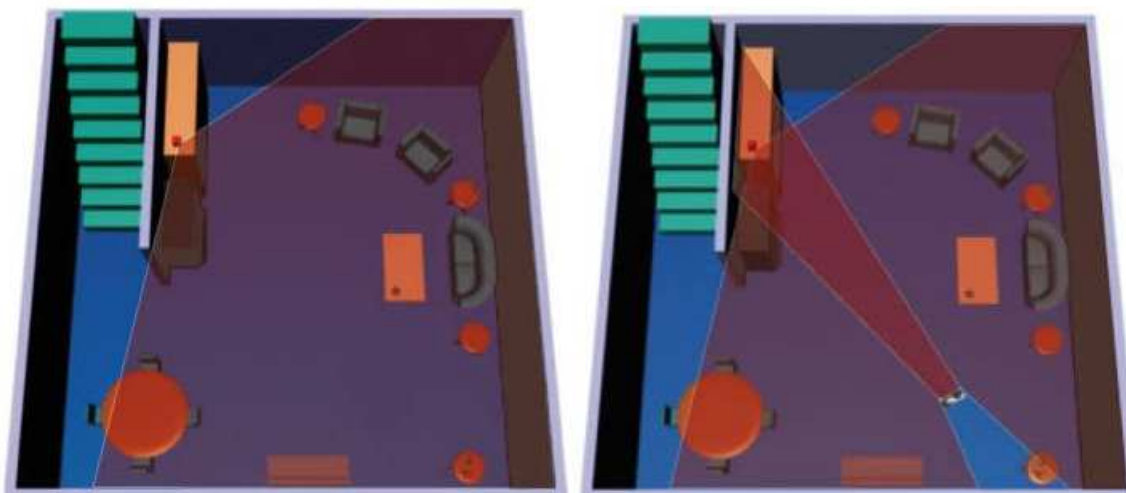


Figura 3.21 – Aplicação de detecção por reflexão.

(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 3.22 mostra este tipo de aplicação (THOMAZINI, 2005; URBANO BRAGA, 2005).

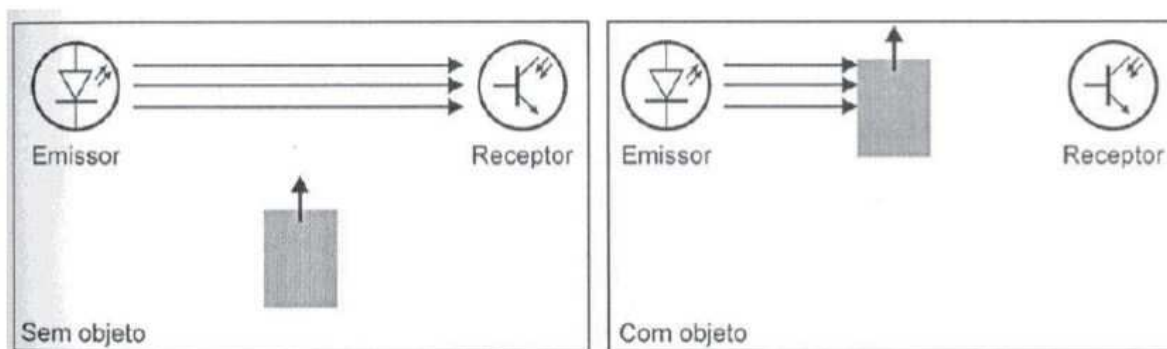


Figura 3.22 - Detecção por interrupção de feixe.

(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 (MIGUEL, 2009). A Figura 3.23 ilustra a aplicação de detecção de entrada e saída de pessoas.



Figura 3.23 – Aplicação de detecção por interrupção de feixe.
(FONTE: REDETEC, 2011).

3.5 – Circuito Infravermelho

Para detecção de vaga foi desenvolvido um circuito utilizando o sensor de infravermelho em cada vaga. Este sensor utiliza a detecção por interrupção de sinal. A figura 3.24 ilustra o circuito emissor em três dimensões. Para cada quatro LEDs há um resistor de 330Ω , pois este circuito é alimentado com 5V.

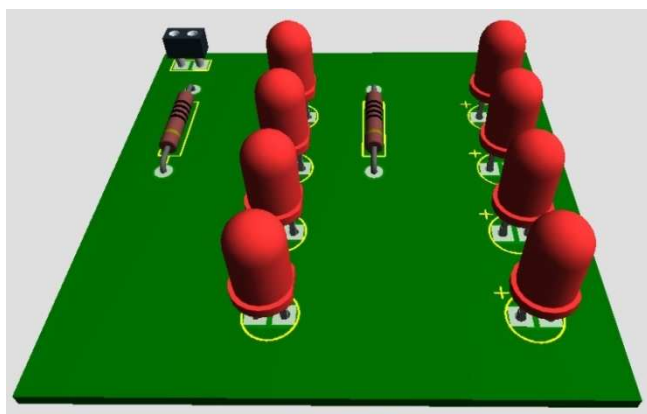


Figura 3.24 – Circuito infravermelho.

Na Figura 3.25 é apresentado o diagrama esquemático do circuito receptor infravermelho.

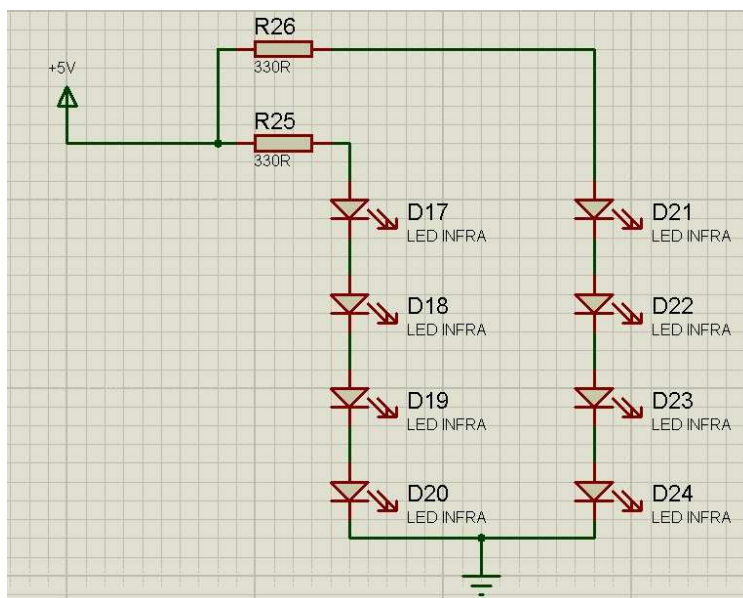


Figura 3.25 – Circuito receptor infravermelho.

3.6 - Circuitos de Inversão para os LEDs

Para a sinalização da vacância ou não das vagas foram utilizados LEDs. O LED vermelho foi usado para indicar que a vaga esta preenchida e o LED verde para indicar que a vaga esta livre. A Figura 3.26 ilustra o circuito feito para o funcionamento desses LEDs de sinalização.

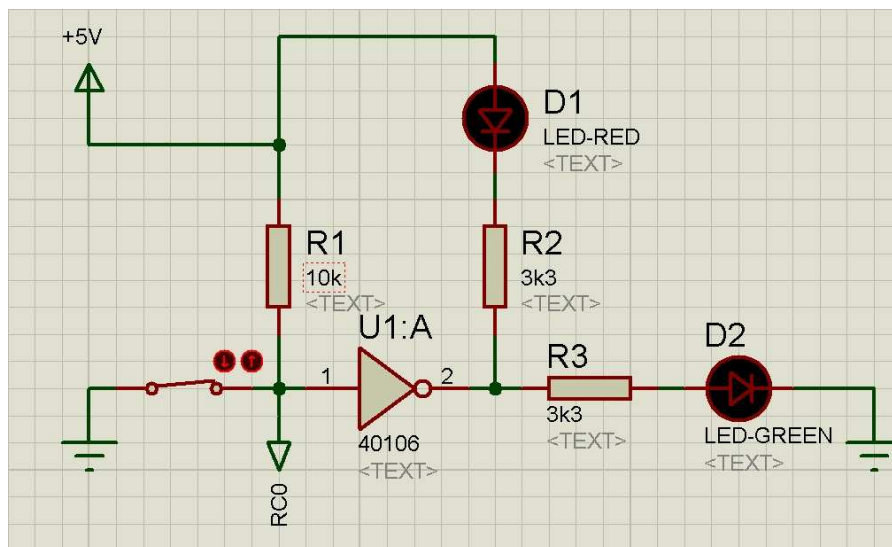


Figura 3.26 – Circuito de inversão.

Este circuito foi nominado de circuito inversor, pois utiliza o chip inversor 40106 da linha CMOS. Este chip inverte o nível lógico de entrada. A Figura 3.26 ilustra uma porta que representa os sinais vindos dos circuitos de infravermelhos.

- Quando a porta esta fechada, representa que o infravermelho detectou presença de algum veículo.
- Quando a porta esta aberta, representa que o infravermelho não detectou presença.

O circuito é alimentado por 5V. O resistor R1 de 10k Ω possui dupla finalidade: polarizar o LED receptor (Rx) infravermelho e manter o nível lógico alto na entrada da porta inversora. Os LEDs também estão representados na Figura 3.26. Os resistores R2 e R3 são necessários para polarizar os LEDs. Estes resistores possuem o valor de 3,3k Ω . Este valor foi calculado pela Lei de Ohm:

$$R = V / I_{LED} = 5V - 1,7V / 10 * 10^{-3} = 330 \text{ Ohm}$$

Onde: 1,7 é a tensão V_{LED} de polarização do LED, ou seja, a tensão minima que ele precisa para funcionar. E I_{LED} é a corrente do LED.

Simulação do circuito:

- Quando a porta esta fechada: o chip inversor recebe nível lógico baixo, este chip inverte este nível polarizando o terminal de catodo do LED vermelho. Este LED acende e o LED verde permanece desligado.
- Quando a porta está aberta: o chip inversor recebe nível lógico alto, este chip inverte este nível polarizando o terminal de anodo do LED verde. Este LED acende e o LED vermelho permanece desligado.

CAPÍTULO 4- TESTES E IMPLEMENTAÇÃO

Este capítulo descreve a implementação e testes durante o desenvolvimento do projeto além da simulação do funcionamento. O item 4.1 é a apresentação do desenvolvimento do projeto. O item 4.2 representa sua topologia. O item 4.3 e seus subitens tratam da descrição dos testes, da montagem da maquete e da programação do PIC. O item 4.4 é a apresentação do funcionamento da maquete.

4.1 – Desenvolvimento do Projeto.

O projeto foi realizado em quatro etapas. A primeira etapa foi o planejamento do projeto e seu estudo bibliográfico. A segunda etapa foi marcada pelo inicio da redação da monografia e pela compras de materiais necessários para a montagem da maquete. A terceira etapa se iniciou com os primeiros testes, pela montagem da maquete e pela programação do PIC. Por fim, a quarta etapa foi a redação final desta monografia. A figura 4.1 mostra essas etapas do desenvolvimento do projeto.

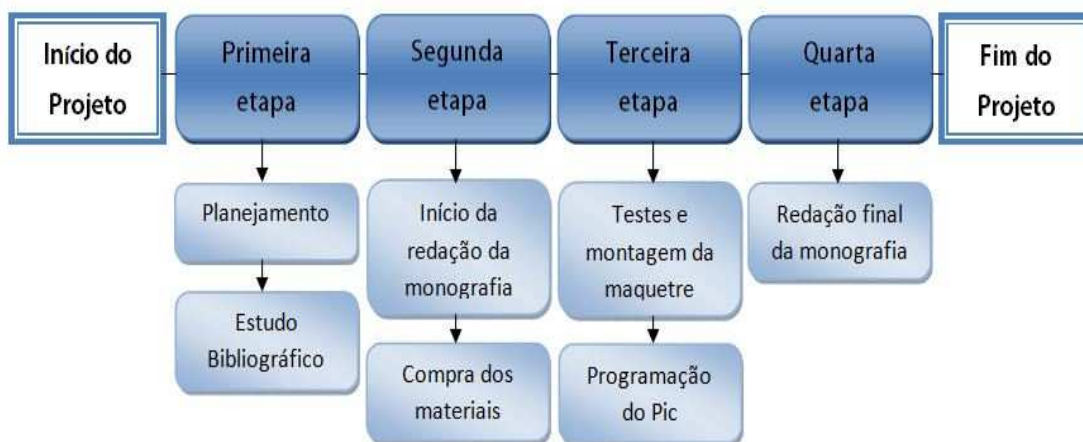


Figura 4.1 – Etapas do desenvolvimento do projeto.

4.2 – Topologia do Projeto

Para o melhor entendimento dos objetivos propostos neste projeto, uma topologia foi ilustrada a fim de facilitar o entendimento da lógica do funcionamento geral do projeto, como mostrado na Figura 4.2.

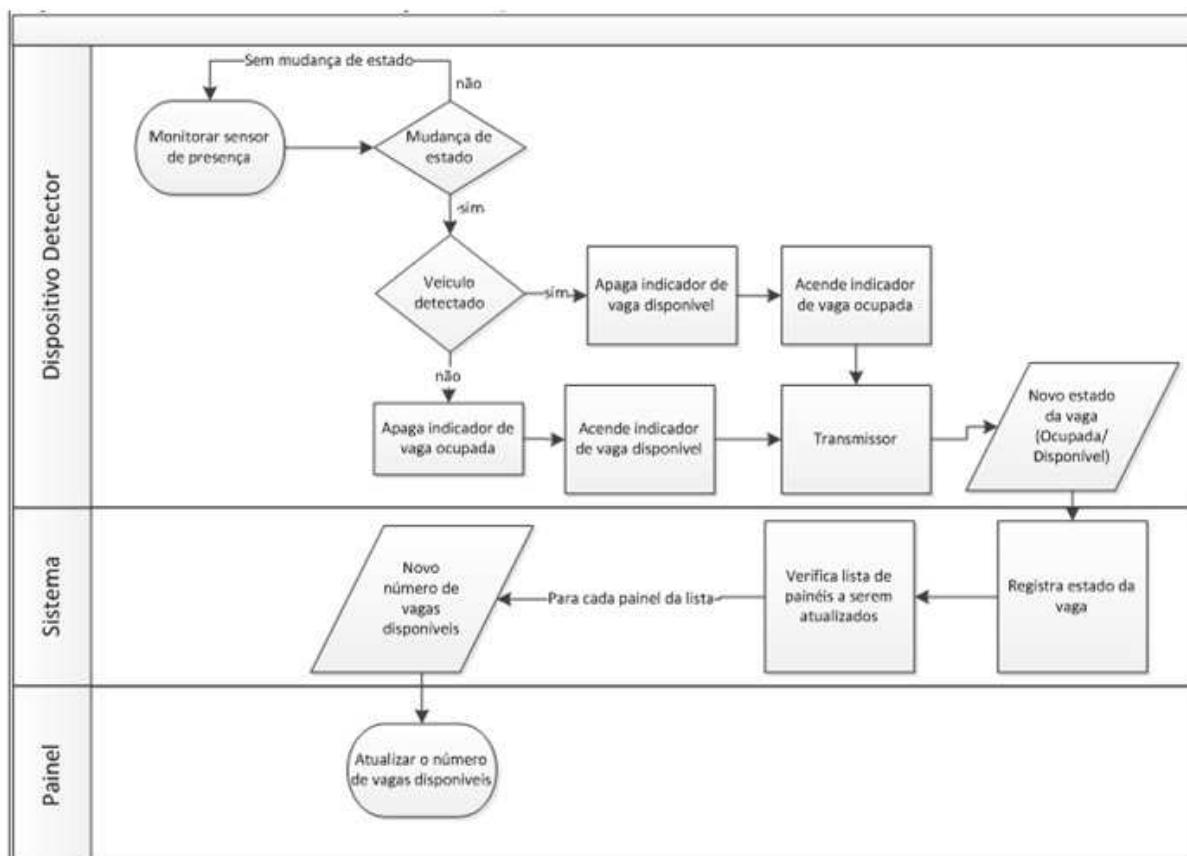


Figura 4.2 – Topologia do projeto.

Conforme ilustrado na figura 4.2 existem dois estados possíveis para o sistema: um estado é quando o detector de presença detecta a presença de um carro na vaga e o outro estado é quando não há carro na vaga. Lembrando que, em cada uma das oito vagas construídas na maquete há um sensor de presença.

O monitor de presença irá sempre monitorar se há alguma mudança no estado da vaga. No primeiro estado este monitor irá detectar a presença de um carro. Detectada a presença irá ser apagado o indicador de vaga disponível e irá acender a luz do indicador de vaga ocupada. Depois disso, é transmitido ao sistema o seu novo estado de vaga ocupada. O sistema, então, registrará este novo estado, analisará quais painéis precisam ser atualizados e informará para cada painel o número que ele deverá mostrar.

No segundo estado o detector de presença não detecta nenhum veículo. Neste caso então, será ligado o indicador de luz disponível e apagada o luz de ocupado. Este novo estado será transmitido para o sistema. O sistema irá registrar

este novo estado, analisará quais painéis precisam ser atualizados e informará para cada painel o número que ele deverá mostrar.

Ao final desses dois estados é possível verificar nos painéis a quantidade de vagas disponíveis no estacionamento.

4.3 – Descrição da Implementação

A implementação do projeto foi dividida em duas fases. A primeira se iniciou com alguns testes dos circuitos e componentes utilizados no projeto e em seguida foi finalizada com a montagem do protótipo. Já a segunda fase foi marcada pela programação do PIC e integração da parte física do projeto com a parte lógica.

4.3.1 – Testes e Montagem da Maquete

A segunda etapa do desenvolvimento do projeto, assim como explicado no item 4.1 deste capítulo, foi a compra dos materiais, dentre eles, o inversor digital 40106, LEDs vermelhos e verdes, infra emissor e receptor, bateria de 3 Volts, suporte para quatro pilhas 1,5 Volts, regulador de tensão de 6 Volts para 5 Volts, uma protoBord, duas resistências de $3k3\Omega$ e uma $10k\Omega$.

Após a compra dos componentes necessários foi realizado um estudo detalhado sobre o funcionamento das entradas e ligações do protobord. Em seguida ocorreu a montagem dos componentes eletrônicos nesta placa.

Assim, pode-se dar início a fase de testes. O primeiro teste realizado foi o do inversor digital 40106. Foi constatado que o material estava funcionando adequadamente, ou seja, quando o receptor do infravermelho não recebe um sinal o LED vermelho está ligado e quando está recebendo acende o LED verde.

Posteriormente foi analisado o brilho desses LEDs, para a constatação se a luz emitida era fraca ou forte. Assim, pode-se definir a resistência mais adequada. Após estes testes, chegou-se ao valor de que a $3k3\Omega$. A Figura 4.3 é uma foto deste teste descrito acima.

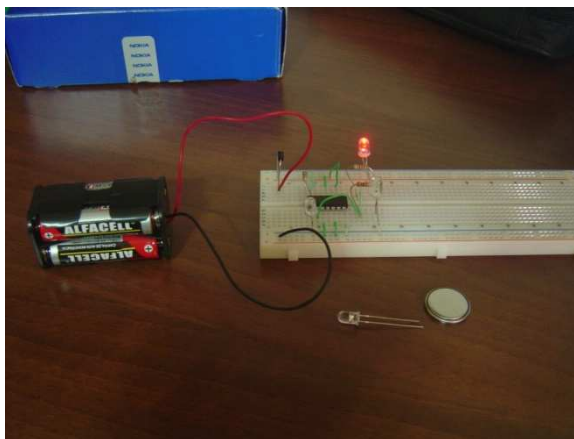


Figura 4.3 – Teste do brilho dos LEDs.

Após estes testes, foi utilizado um programa para projetos e simulações de circuitos eletrônicos, chamado Proteus. Este é um software de simulação de circuitos Proteus ISIS Schematic Capture® da Labcenter Electronics® na versão 7.7. Ele é um simulador de microprocessadores, captura esquemática, e placa de circuito impresso (PCB design). É desenvolvido pela empresa inglesa Labcenter Electronics. Este programa possibilita verificar o correto funcionamento dos circuitos.

Com Proteus foi encontrado um problema em relação à simulação de uma vaga de estacionamento, uma vez que no programa de desenhos de circuitos eletrônicos não tem como simular um infravermelho receptor e emissor. Sendo assim, esta simulação foi feita por meio de um interruptor. Quando este estiver aberto, representa uma vaga ocupada, e quando fechado, representa uma vaga disponível. Na Figura 4.4 é representado este interruptor aberto e a Figura 4.5 o representa fechado.

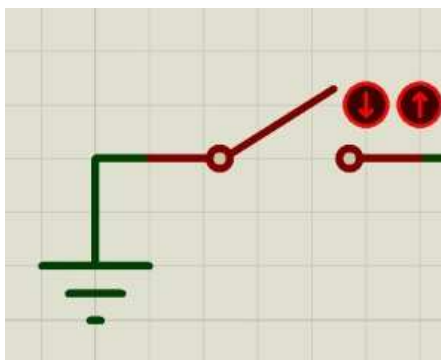


Figura 4.4 – Interruptor de simulação aberto.



Figura 4.5 – Interruptor de simulação fechado.

Depois da experiência com o interruptor foi simulada, neste mesmo programa, a disponibilidade ou não da vaga de estacionamento utilizando o circuito inversor já explicado no capítulo anterior. O mesmo circuito foi utilizado para as outras 7 vagas construídas no projeto. A Figura 4.6 ilustra o circuito projetado.

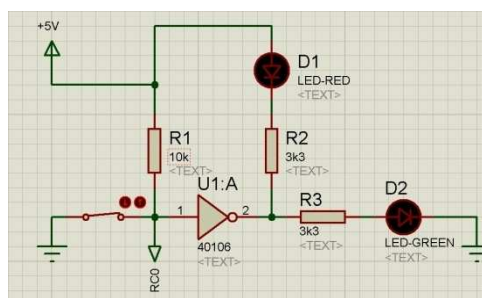


Figura 4.6 – Simulação do circuito inversor.

O próximo teste foi a simulação do *display* de 7 segmentos utilizado para poder mostrar a quantidade das vagas disponíveis no estacionamento. Foram incluídos dois *displays* para informar a quantidade de vagas disponíveis em cada setor A e B e um terceiro *display* seria a soma dos dois setores, ou seja A e B, para colocar na entrada do estacionamento, com a informação exata sobre a disponibilidade das vagas. A figura 4.7 mostra a simulação destes três *displays*.

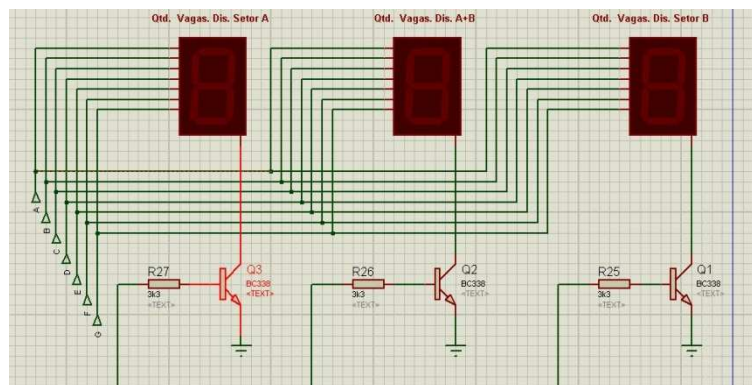


Figura 4.7 – Simulação dos três *displays*.

Por fim, para controlar todos os dispositivos do projeto, foi incluído um microcontrolador PIC16F876A. Além de controlar os dispositivos, o microcontrolador PIC16F876A também envia um histórico para o computador todas as vezes que houver mudança no status da vaga.

Assim que foi constatado que tudo estava funcionando conforme o esperado, pode-se começar a confecção da placa de circuitos impressa, no programa Proteus, conforme demonstrado na figura 4.8.

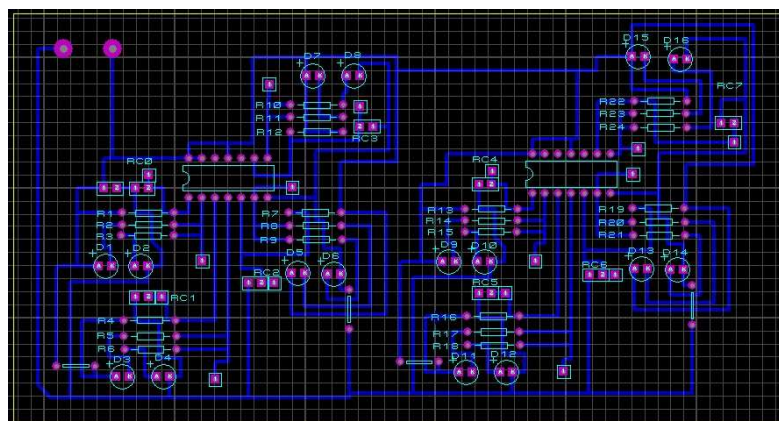


Figura 4.8 - Simulação da placa de circuitos no programa Proteus.

Como a confecção da placa de circuitos no programa Proteus foi realizada com sucesso, somente faltava a simulação de uma nova placa do circuito de infravermelho emissor que também foi simulado no Proteus. A figura 4.9 ilustra a simulação deste circuito.

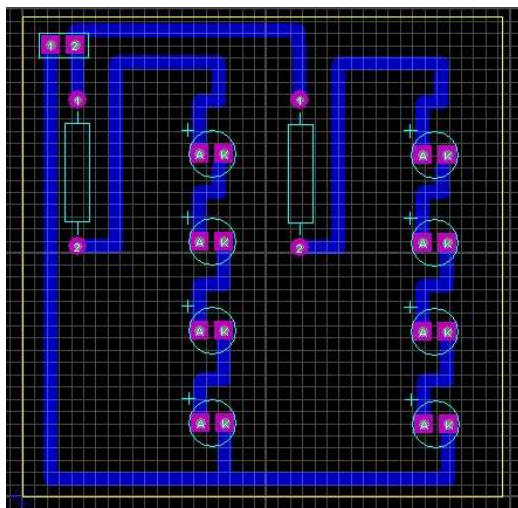


Figura 4.9 – Simulação do circuito infravermelho.

Assim que todos os circuitos estavam desenhados, iniciou-se a construção das placas.

O primeiro passo para a construção da placa foi imprimir o desenho do circuito em uma transparência.

A placa de circuito estava com gordura e sujeira, tendo a necessidade de ser limpar, passando uma esponja de aço, para retirar os resquícios encontrados e dar mais brilho para a mesma. A Figura 4.10x ilustra esta higienização.



Figura 4.10 – Higienização da placa.

Depois da higienização foi colada à placa a transparência contendo o circuito. A figura 4.11 mostra esta etapa.



Figura 4.11 – Colagem da transparência à placa.

Para que a tinta da transparência pudesse passar para a placa foi necessário envolver a placa em um tecido e então utilizar um ferro quente. A figura 4.12 ilustra uma foto deste processo.



Figura 4.12 – Passagem da tinta da transparência para a placa.

Em seguida foi retirada a transparência da placa e, então, a placa foi mergulhada no perclorato de ferro para corroer o cobre das partes não pintadas. A figura 4.13 ilustra este processo de corrosão.

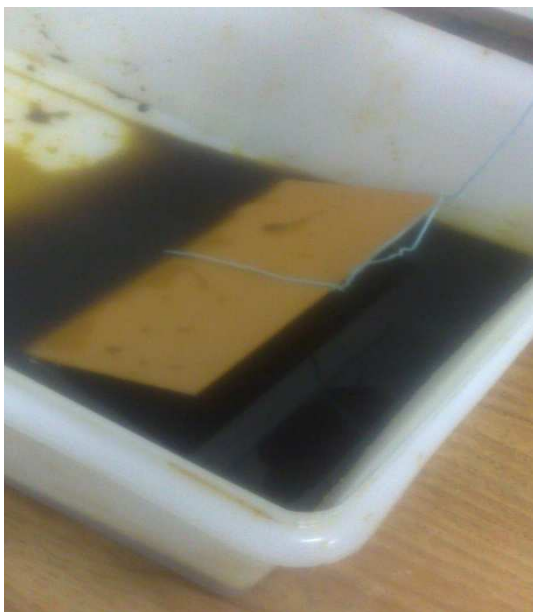


Figura 4.13 – Processo de corrosão da placa.

Após esperar o tempo necessário para a corrosão, a parte pintada permanece com o cobre permitindo o funcionamento do circuito. A Figura 4.14 ilustra a placa sendo retirada da substância e o circuito desenhado com perfeição.

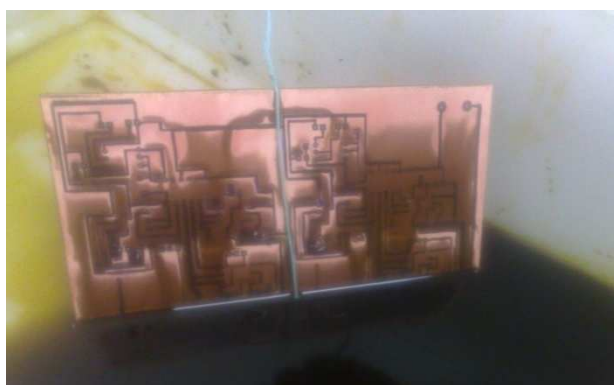


Figura 4.14 – Circuito desenhado na placa.

Depois de retirada a placa de circuito da substância química, removeu-se a tinta restante com a esponja de aço como mostra a Figura 4.15.



Figura 4.15 - Remoção da tinta restante.

Depois da higienização total da placa de circuito, restando apenas as trilhas do circuito, perfurou-se a placa para encaixar os componentes eletrônicos, conforme demonstrado da Figura 4.16:



Figura 4.16 – Perfuração da placa.

O Último passo foi soldar os componentes cada um em seu devido lugar. A Figura 4.17 ilustra os componentes sendo soldados.



Figura 4.17 - Componentes sendo soldados.

Este processo foi utilizado em todas as placas do projeto. Uma placa pronta pode ser observada na figura 4.18.

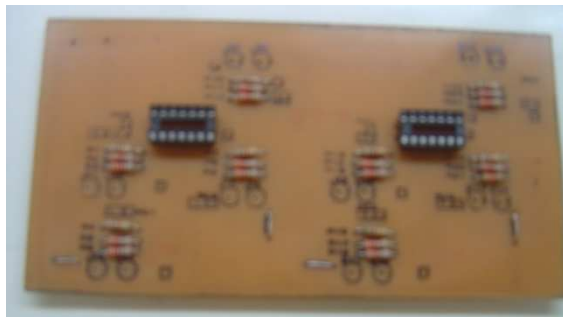


Figura 4.18 – Placa pronta.

Com os testes concluídos com sucesso e com as placas prontas a montagem da maquete foi iniciada. Como pode-se notar na figura 4.19 foram conectados o kit de desenvolvimento, as placas construídas e os LEDs de sinalização.

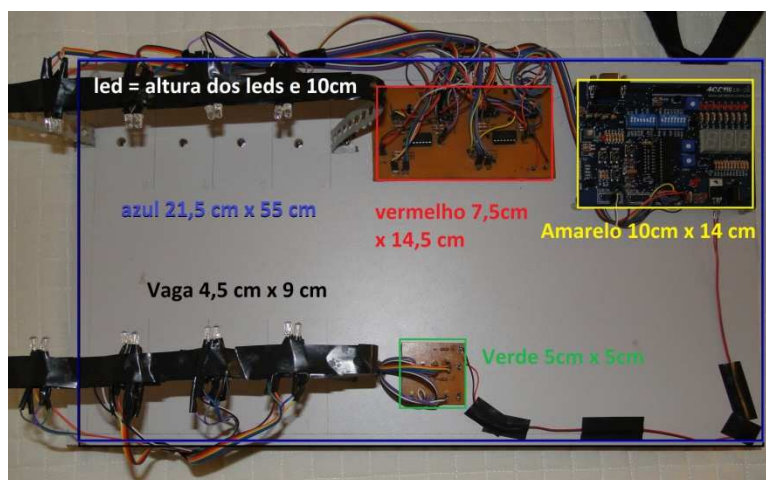


Figura 4.19 – Maquete montada.

4.3.2 – Programação de PICs em C

Com a maquete pronta iniciou-se uma nova etapa, com o objetivo de programar uma rotina na linguagem c. Para isso foi utilizado o programa MikroC Pro. O MikroC Pro é um compilador de linguagem C para PIC, o único que reúne todo o poder e a flexibilidade proporcionada pelo ANSI C, com o IDE mais avançado do mercado. Este compilador possui exemplos práticos e uma documentação completa

para permitir um desenvolvimento rápido na programação de microcontroladores PIC. Extremamente indicado para iniciantes e profissionais que desejam facilidade e velocidade para desenvolver projetos utilizando este tipo de MCU.

Após o término da programação foi compilado e gerado várias extensões de arquivos como, por exemplo, .c .hex .dlt .dbg .mcpqi, como mostra a figura 4.20.



Figura 4.20 – Extensões dos arquivos gerados.

Dentre as extensões de arquivo utilizou-se o .hex para gravar no microcontrolador PIC16F876A.

O código utilizado para o projeto está especificado e explicado no Apêndice A.

4.4 – Apresentação do funcionamento da maquete

Após todos os dispositivos devidamente testados a maquete foi construída. Ela possui comprimento de 21,5 cm e largura de 55cm. Na Figura 4.21 é apresentada uma foto desta maquete finalizada.

Na placa pintada de vermelho na foto estão conectados: todos os LEDs (vermelhos e verdes) e todos os receptores de infravermelho. No total são oito LEDs vermelhos, oito verdes e oito receptores. Nesta placa em vermelho também estão presentes os dois circuitos conversores digitais. Cada circuito conversor é

responsável por quatro LEDs vermelhos, quatro LEDs verdes e quatro receptores infravermelhos.

A placa pintada de amarelo na foto é a placa do kit de desenvolvimento ACEPIC16F876A. Nele estão acoplados os dispositivos: microcontrolador, regulador de tensão, *display* de 7 segmentos e o conector serial DB9.

Na placa pintada em verde está o circuito emissor infravermelho.

As vagas construídas na maquete possuem 9 cm de comprimento e 4,5 de largura e , estão localizadas do lado esquerdo da Figura 4.21. Em cada vaga há um LED emissor, um LED receptor e dois LEDs sinalizadores (verde e vermelho).

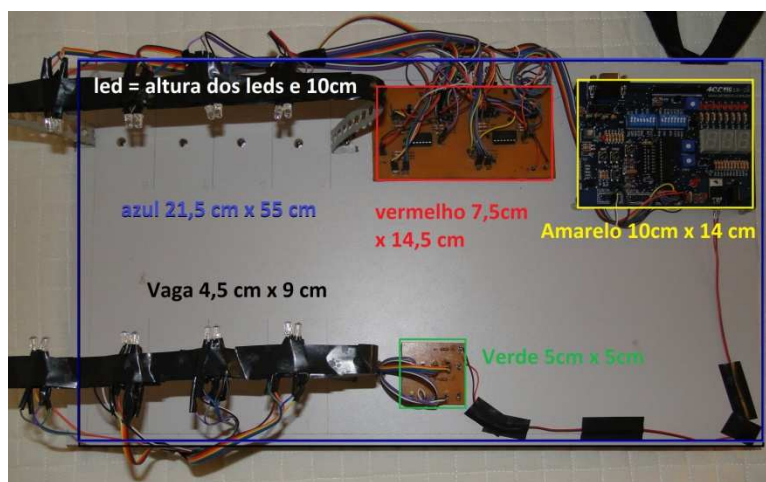


Figura 4.21 – Maquete finalizada.

Com a montagem de toda a estrutura da maquete, a simulação está pronta para ser executada.

CAPÍTULO 5 – CONCLUSÃO

O projeto foi desenvolvido com o intuito de melhorar a qualidade de vida dos condutores de veículos, que perdem muito tempo á procura de vagas. Para isto, foi desenvolvido um protótipo em escala reduzida de um estacionamento automatizado utilizando um microcontrolador. Esse projeto permite ao usuário saber previamente onde estacionar logo ao entrar no estacionamento . Para que este projeto seja utilizado em padrões reais seria necessário adequá-lo à realidade trocando os materiais utilizados de baixa escala.

O objetivo geral deste trabalho é facilitar a procura por vagas. Há também outra vantagem: com menos tempo gasto com a procura de vagas, menos tempo o motor do carro fica ligado, assim conseguintemente, contribui com a diminuição da poluição do meio ambiente além dos engarrafamentos dentro dos estacionamentos nos horários de maior fluxo.

Testes foram realizados com o protótipo montado e foi observado que em todos os casos testados o protótipo funcionou adequadamente. Concluindo então, que o projeto desenvolvido teria boas chances de funcionalidade para o cotidiano dos motoristas. O controle de vagas em um estacionamento trará maior agilidade para quem quer estacionar, além dos aspectos ambientais como já tratado anteriormente.

As tarefas propostas para a realização deste trabalho foram em uma maquete. O programa em C proposto para controlar o sistema foi compilado e executado; O PIC para gerenciar a parte lógica do projeto foi utilizado; Os sensores infravermelhos para informar a quantidade de vagas disponíveis foram construídos. LEDs foram utilizados para sinalizar o *status* da vaga; E o circuito para controlar todas as vagas e LEDs também foi construído. Alcançando assim, todos os objetivos e tarefas propostas para este projeto.

Os resultados obtidos com o protótipo foram satisfatórios para os objetivos iniciais. Porém este trabalho se limita a isto não considerando assim outros aspectos de segurança.

5.1 – Sugestões de Trabalhos Futuros

Como sugestão para outros projetos nesta área, recomenda-se o controle da entrada e da saída dos veículos, podendo cobrar pelo tempo estacionado na vaga. É possível também, implementar o mesmo projeto usando a tecnologia de transferência de informações através do dispositivo Zigbee, substituindo os fios. Por meio da utilização de um display de LCD, pode-se mostrar ao condutor o horário de entrada e saída do seu veículo, a data específica, o nível de temperatura ambiente, e o próximo número indicado para a localização da vaga a ser ocupada no estacionamento.

REFERÊNCIAS BIBLIOGRÁFIA

ACEPIC, (2011) – Datasheet ACEPIC_18_28.

Acesso em: 10/03/2011.

Em: www.acepic.com.br/manuais/acepic_18_28.pdf

BUDAPEST PARKING, (2010) - Estacionamento inteligente de Budapeste.

Acesso em: 24/03/2011.

Em: <http://www.noticiasautomotivas.com.br/video-estacionamento-inteligente-de-budapeste/>

Camargo & Neves, (2002) – Edifício Inteligente.

Acesso em: 14/02/2011.

Em: <http://sites.google.com/site/edificioseedificio/edificios-inteligentes>

CARROS ANTIGOS, (2011) – Carros Antigos.

Acesso em: 23/03/2011.

Em: http://www.carroantigo.com/portugues/conteudo/curio_hist_carro_brasileiro.htm

CIRCUIT SHOP. (2011) - MAX232.

Acesso em: 12/04/2011.

Em: http://circuitshop.com.br/loja/popup_image.php?plD=22

CORREIO BRAZILIENSE, (2001) - Brasilienses investem cada vez mais na compra de veículos.

Acesso em: 17/03/2011.

Em: http://www.correiobraziliense.com.br/app/noticia/cidades/2011/02/13/interna_cidade/df,237498/brasilienses-investem-cada-vez-mais-na-compra-de-veiculos.shtml

DETRAN-DF, (2011) - Departamento de Trânsito do Distrito Federal. Sessão Estatísticas. .

Acesso em: 20/05/2011.

Em: <http://www.detran.df.gov.br/>

Estadão de São Paulo, (2010) – Opinião.

Acesso em: 15/03/2011.

Em: http://www.estadao.com.br/estadaodehoje/20110311/not_imp690346,0.php

GOMES, Laurentino – 1808.

1º Edição. Editora Planeta do Brasil. Brasil, 2006.

414 páginas.

KIOSKEA, (2011) – Imagem conector DB9.

Acesso em: 12/05/2011.

Em: <http://pt.kioskea.net>

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

Acesso em: 26/04/2011.

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

METALICA, (2011) – Estacionamentos Verticais.

Acesso em: 10/04/2011.

Em: <http://www.metalica.com.br/o-que-sao-e-como-funcionam-os-estacionamentos-verticais>

MIGUEL, Afonso. Sensor Infravermelho.

Acesso em: 12/01/2011.

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

MUSEU HISTÓRICO NACIONAL, (2011).

Acesso em: 15/02/2011.

Em: <http://www.museuhistoriconacional.com.br/mh-e-330i.htm>

PRADA, (2010) – Estacionamento Morumbi Shopping.

Acesso em: 25/05/2011

Em: <http://fotos.estadao.com.br/cidades-novo-sistema-de-indicacao-de-vagas-livres-no-estacionamento-do-shopping-morumbi-em-sao-paulo.galeria,,116789,,9,,0.htm>

PUCRS, (2011) - Imagem display 7 segmentos.

Acesso em: 12/05/2011.

Em: http://www.ee.pucrs.br/~terroso/html/7_segmentos.html

RADCOMALARMES, (2011) - Imagem infravermelho detecção reflexão.

Acesso em: 15/05/2011.

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

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

Acesso em: 15/05/2011.

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

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

Acesso em: 21/04/2011.

Em: http://www.machine-informationssystem.com/RS232_Pinouts.html

SCHOR (1999) – Revolução Invisível.

Acessado em: 15/03/2011.

Em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-88391999000300014

SOMNUSCARS, (2009) – Para Evitar uma Vaga Ocupada.

Acesso em: 14/02/2011.

Em: <http://somniaucars.com/2009/07/21/para-evitar-uma-vaga-ocupada/>

SWEET, (2011) – Imagem conversão binária.

Acesso em: 21/05/2011.

Em: http://sweet.ua.pt/~a20042/doc/guiao_aluno.pdf

TEXAS INSTRUMENTS, (2004) - MAX232. Datasheet do MAX232.

Acesso em: 25/02/2011.

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

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

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

3ª edição. Editora Bookman. São Paulo 2008.

PEREIRA, Fabio. Programação em C.

7ª edição. Editora Érica.

MARKUS, Otávio. Circuitos Elétricos.

8ª edição. Editora Érica.

APÊNDICE A

```

//Portas para habilitação dos displays
#define dsp1 porta.f1 //Display ligado na porta A no pino 1
#define dsp2 porta.f2 //Display ligado na porta A no pino 2
#define dsp3 porta.f3 //Display ligado na porta A no pino 1

//Sensores do SETOR A
#define s1 portc.f0 //Sensor ligado na porta C no pino 0
#define s2 portc.f1 //Sensor ligado na porta C no pino 1
#define s3 portc.f2 //Sensor ligado na porta C no pino 2
#define s4 portc.f3 //Sensor ligado na porta C no pino 3

//Sensores do SETOR B
#define s5 portc.f4 //Sensor ligado na porta C no pino 4
#define s6 portc.f5 //Sensor ligado na porta C no pino 5
#define s7 porta.f4 //Sensor ligado na porta C no pino 6
#define s8 porta.f5 //Sensor ligado na porta C no pino 7

//defini variaveis booleanas só pra facilitar a visualização.
#define true 1
#define false 0

//Configuração dos terminais do display CATODO-COMUM em hexadecimal
const num0 = 0x3F; //Numero 0
const num1 = 0x06; //Numero 1
const num2 = 0x5B; //Numero 2
const num3 = 0x4F; //Numero 3
const num4 = 0x66; //Numero 4
const num5 = 0x6D; //Numero 5
const num6 = 0x7C; //Numero 6
const num7 = 0x07; //Numero 7
const num8 = 0x7F; //Numero 8
const num9 = 0x6F; //Numero 9

//declaração de variáveis e matriz do tipo INTEIRA
int setora = 0, setorb = 0, soma = 0, sensor[] = {0,0,0,0,0,0,0,0};
int disp = 0;
int l;
//Inicio do prototipo de funções
void verifica_sensores();
void escreve_log();
void seta_display();

void main()
{
    TXSTA.TXEN = 1; //Habilita transmissão da USART
    TXSTA.BRGH = 1; //Baud rate Alto
    RCSTA.SPEN = 1; //Habilitação da USART..
    PIE1.RCIE = 1; //Habilitação da interrupção de recepção da USART.. NÃO USADO

    CMCON = 7; //Desabilita comparadores, libera port A0, 1, 2 e 3..
    //Inicio da configuração dos A/D's
    ADCON1.PCFG3 = 0;
    ADCON1.PCFG2 = 1;
    ADCON1.PCFG1 = 1;
    ADCON1.PCFG0 = 1;

```

```

//Configuração da direção das portas.. entrada/saída
TRISB = 0b10000000; // 0 saidas 1 entradas
PORTB = 0; // todas as pinos da porta B no nivel zero volts
TRISA = 0b110000;
PORTA = 0;
TRISC = 0b10111111;
PORTC = 0;

//Definição da velocidade de comunicação da USART, 9600 bits por segundo
UART1_Init(9600);

//Loop permanente
while(true)
{
    //Chama as funções
    verifica_sensores();

    for(dis=0; dis<2;dis++){
        seta_display();
    }
}

}

void verifica_sensores(){
//verdadeiro => ocupado

    if(s1!=sensor[0] || s2!=sensor[1] || s3!=sensor[2]
        || s4!=sensor[3] || s5!=sensor[4] || s6!=sensor[5] || s7!=sensor[6] || s8!=sensor[7] )
    {
        escreve_log();
    }
}

void escreve_log()
{
    setora=0;
    setorb=0;

    //verifica se os sensores ESTÃO ativos.
    UART1_Write_Text(" ");
    if(s1)
    {
        sensor[0] = 1;
        setora++;
        UART1_Write_Text("OA1"); // Enviar o log para o pc vaga A1 Ocupada
    } else
    {
        sensor[0] = 0;
        UART1_Write_Text("DA1"); // Enviar o log para o pc vaga A1 Disponivel
    }
}

if(s2)
{
    sensor[1] = 1;
    setora++;
    UART1_Write_Text("OA2"); // Enviar o log para o pc vaga A2 Ocupada
} else

```

```

{
    sensor[1] = 0;
    UART1_Write_Text("DA2"); // Enviar o log para o pc vaga A2 Disponivel
}
if(s3)
{
    sensor[2] = 1;
    setora++;
    UART1_Write_Text("OA3"); // Enviar o log para o pc vaga A3 Ocupada
} else
{
    sensor[2] = 0;
    UART1_Write_Text("DA3"); // Enviar o log para o pc vaga A3 Disponivel
}
if(s4)
{
    sensor[3] = 1;
    setora++;
    UART1_Write_Text("OA4"); // Enviar o log para o pc vaga A4 Ocupada
} else
{
    sensor[3] = 0;
    UART1_Write_Text("DA4"); // Enviar o log para o pc vaga A3 Disponivel
}
if(s5)
{
    sensor[4] = 1;
    setorb++;
    UART1_Write_Text("OB1"); // Enviar o log para o pc vaga B1 Ocupada
} else
{
    sensor[4] = 0;
    UART1_Write_Text("DB1"); // Enviar o log para o pc vaga B1 Disponivel
}
if(s6)
{
    sensor[5] = 1;
    setorb++;
    UART1_Write_Text("OB2"); // Enviar o log para o pc vaga B2 Ocupada
} else
{
    sensor[5] = 0;
    UART1_Write_Text("DB2"); // Enviar o log para o pc vaga B2 Disponivel
}
if(s7)
{
    sensor[6] = 1;
    setorb++;
    UART1_Write_Text("OB3"); // Enviar o log para o pc vaga B3 Ocupada
} else
{
    sensor[6] = 0;
    UART1_Write_Text("DB3"); // Enviar o log para o pc vaga B3 Disponivel
}
if(s8)
{
    sensor[7] = 1;
    setorb++;
    UART1_Write_Text("OB4"); // Enviar o log para o pc vaga B4 Ocupada
} else

```



```

{
    sensor[7] = 0;
    UART1_Write_Text("DB4"); // Enviar o log para o pc vaga B4 Disponivel
}

soma= setora + setorb;

//Faz a verificação das quantidades de vagas disponiveis do setor A
switch(4 - setora)
{
    case 0:
        UART1_Write_Text("AA0"); // Enviar numero de vags disponivel no
        setor A e 0
        break;
    case 1:
        UART1_Write_Text("AA1"); // Enviar numero de vags disponivel no
        setor A e 1
        break;
    case 2:
        UART1_Write_Text("AA2"); // Enviar numero de vags disponivel no
        setor A e 2
        break;
    case 3:
        UART1_Write_Text("AA3"); // Enviar numero de vags disponivel no
        setor A e 3
        break;
    case 4:
        UART1_Write_Text("AA4"); // Enviar numero de vags disponivel no
        setor A e 4
        break;
}
//Faz a verificação das quantidades de vagas disponiveis do setor B
switch(4 - setorb)
{
    case 0:
        UART1_Write_Text("BB0"); // Enviar numero de vags disponivel no
        setor B e 0
        break;
    case 1:
        UART1_Write_Text("BB1"); // Enviar numero de vags disponivel no s
        setor B e 1
        break;
    case 2:
        UART1_Write_Text("BB2"); // Enviar numero de vags disponivel no
        setor B e 2
        break;
    case 3:
        UART1_Write_Text("BB3"); // Enviar numero de vags disponivel no
        setor B e 3
        break;
    case 4:
        UART1_Write_Text("BB4"); // Enviar numero de vags disponivel no
        setor B e 4
        break;
}
//Faz a diferença entre o numero total e a soma das vagas disponiveis.
switch(8 - soma)
{
    case 0:
        UART1_Write_Text("SS0"); // Enviar numero de vags disponivel no

```

```

        estacionamento e 0
        break;
    case 1:
        UART1_Write_Text("SS1"); // Enviar numero de vagas disponivel no
        estacionamento e 1
        break;
    case 2:
        UART1_Write_Text("SS2"); // Enviar numero de vagas disponivel no
        estacionamento e 2
        break;
    case 3:
        UART1_Write_Text("SS3"); // Enviar numero de vagas disponivel no
        estacionamento e 3
        break;
    case 4:
        UART1_Write_Text("SS4"); // Enviar numero de vagas disponivel no
        estacionamento e 4
        break;
    case 5:
        UART1_Write_Text("SS5"); // Enviar numero de vagas disponivel no
        estacionamento e 5
        break;
    case 6:
        UART1_Write_Text("SS6"); // Enviar numero de vagas disponivel no
        estacionamento e 6
        break;
    case 7:
        UART1_Write_Text("SS7"); // Enviar numero de vagas disponivel no
        estacionamento e 7
        break;
    case 8:
        UART1_Write_Text("SS8"); // Enviar numero de vagas disponivel no
        estacionamento e 8
        break;
    }
    UART1_Write_Text("\n\r");
}
void seta_display()
{
    //faz a varredura dos dados na matriz sensor[] e soma às suas respectivas var
    riaveis de ambiente
    const int tempo_display = 100;

    delay_ms(tempo_display);
    //Habilita o displays do setor A
    dsp1 = true;
    dsp2 = false;
    dsp3 = false;

    //Faz a verificação das vagas disponiveis do setor A
    switch(4 - setora)
    {
        case 0:
            PORTB = num0; //vagas disponiveis do setor A e 0
            break;
        case 1:
            PORTB = num1; //vagas disponiveis do setor A e 1
            break;
        case 2:
            PORTB = num2; //vagas disponiveis do setor A e 2

```

```

        break;
    case 3:
        PORTB = num3; //vagas disponiveis do setor A e 3
        break;
    case 4:
        PORTB = num4; //vagas disponiveis do setor A e 4
        break;
}
delay_ms(tempo_display);
//Habilita o displays do setor B.
dsp1 = false;
dsp2 = false;
dsp3 = true;

//Faz a verificação das vagas disponiveis do setor B
switch(4 - setorb)
{
    case 0:
        PORTB = num0; //vagas disponiveis do setor B e 0
        break;
    case 1:
        PORTB = num1; //vagas disponiveis do setor B e 1
        break;
    case 2:
        PORTB = num2; //vagas disponiveis do setor B e 2
        break;
    case 3:
        PORTB = num3; //vagas disponiveis do setor B e 3
        break;
    case 4:
        PORTB = num4; //vagas disponiveis do setor B e 4
        break;
}

delay_ms(tempo_display);
//Habilita os displays da entrada do estacionamento.
dsp1 = false;
dsp2 = true;
dsp3 = false;

//Faz a diferença entre o numero total e a soma das vagas disponiveis.
switch(8 - soma)
{
    case 0:
        PORTB = num0; //vagas disponiveis no estacionamento e 0
        break;
    case 1:
        PORTB = num1; //vagas disponiveis no estacionamento e 1
        break;
    case 2:
        PORTB = num2; //vagas disponiveis no estacionamento e 2
        break;
    case 3:
        PORTB = num3; //vagas disponiveis no estacionamento e 3
        break;
    case 4:
        PORTB = num4; //vagas disponiveis no estacionamento e 4
        break;
    case 5:
        PORTB = num5; //vagas disponiveis no estacionamento e 5

```

```
        break;
    case 6:
        PORTB = num6; //vagas disponiveis no estacionamento e 6
        break;
    case 7:
        PORTB = num7; //vagas disponiveis no estacionamento e 7
        break;
    case 8:
        PORTB = num8; //vagas disponiveis no estacionamento e 8
        break;
    }
}
```