

**ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA**

**JOÃO PAULO CARVALHO DE FREITAS
MICHEL SAWAYA HIRSCHHEIMER**

**Sistema de Identificação de Linha de Ônibus por
Rádio Freqüência**

SÃO PAULO

2006

João Paulo Caravalho de Freitas
Michel Sawaya Hirschheimer

*nota final
9,2 (nove e dois)*

WJM

**Desenvolvimento de um sistema de identificação de linhas de ônibus por
radio freqüência**

**Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo como trabalho final de formatura**

083

**Área de Concentração:
Engenharia Mecânica com
Ênfase em Automação e Controle**

**Orientador: Prof. Dr.
José Reinaldo Silva**

**São Paulo
2006**

DEDALUS - Acervo - EPMN



31600012472

FICHA CATALOGRAFICA

1574185

Freitas, João Paulo Carvalho de
Sistema de identificação de linhas de ônibus / J.P.C. de Freitas, M.S. Hirschheimer. -- São Paulo, 2006.
p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Identificação (Teoria de sistemas e controle) 2.Transport
coletivo I.Hirschheimer, Michel Sawaya II.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III.t.

Dedicatória

**Dedicamos esse trabalho aos colegas,
professores e familiares que nos
auxiliaram de diversas formas.**

**As pequenas oportunidades são,
Freqüentemente, o início de grandes
empreendimentos.
(Demóstenes)**

RESUMO

Melhorar a vida do usuário do transporte coletivo e das operadoras de ônibus em centros urbanos é a proposta deste projeto. Mais especificamente este trabalho apresenta o desenvolvimento de um sistema de identificação de linhas de ônibus a fim de auxiliar deficientes visuais por meio de avisos sonoros.

Através da análise de requisitos e das restrições existentes, avaliamos a modelagem do problema para utilização da Radio Freqüência como ferramenta de identificação dos veículos. Este sistema baseia-se na tecnologia RF de transferência de dados em que transmissores são instalados nos ônibus e receptores em antenas localizadas nos pontos de ônibus.

A linha identificada é anunciada por alto-falantes auxiliando, assim, tanto os usuários comuns quanto deficientes visuais que, portanto, poderão requisitar paradas sem ajuda de outros usuários. Dando, desta maneira, uma maior independência para tais usuários.

Os transmissores são associados a cada ônibus para que seu número de serie possa ser lido a distância. Desta forma, ao aproximar-se da estação montada no ponto de ônibus provida de antena, pode-se detectar o ônibus através do acesso a um banco de dados com as informações referentes a sua rota.

Neste trabalho esses aspectos são abordados no desenvolvimento de um protótipo que realiza as funções especificadas que possibilite o atendimento ao sistema proposto.

Palavras Chave: ônibus, radio freqüência, deficientes visuais.

ABSTRACT

This work presents a study for the implementation of a structure to help visually impaired people to recognize buses lanes. For that we are using Radio Frequency (RF), which is an automatic identification method, relying on storing and remotely retrieving data using devices called RF modules. Based on Transmitter and Receiver that are able to communicate themselves through electromagnetic waves, those can be also used in moving objects.

With transmitters attached on buses, some sensors installed on the bus stops read the containing information (identification number) of the transmitters passing by. Therefore, software's programmed to recognize the numbers coming from the sensors of the buses are needed to recognize this information. This way a data base installed on processors assembled on the system can announce it through speakers.

Since those sensors (hardware) to assemble this apparatus to work are already commercially available, our study and effort goes mainly to matching the specifications to deal with the structure needed and software development. This software must be able to integrate the signals coming from the RF transmitter, the data base of the buses and the communication through speakers. Using separate structures of command on programming, the processor deals with the signals coming from the RF set to integrate it to the data base and then play the recorded voice information regarding each lane to be communicated.

In addition, due to the ranges and velocities of the buses, to have a clear reading, different kinds of modules and frequencies must be evaluated to get a clear reading. Hence, the alternative that we need to choose can fit the best solution of this matter. Yet the configuration must match some standardization because we must be able to attend the necessary specifications for the homologation of the product on the national agency of telecommunication (ANATEL).

On the whole the project arrangement improves the identification made by users of the buses lanes. While waiting on the bus stop, they can get information about the buses paths that they need to make their journey.

Keywords: bus, radio frequency, visual deficient, wireless.

LISTA DE FIGURAS

<i>Figura 1 – Sistema de monitoramento por GPS</i>	21
<i>Figura 2 - Indução Magnética do TAG pela Antena</i>	24
<i>Figura 3 - Tag</i>	25
<i>Figura 4 – Ondas para transmissão serial</i>	33
<i>Figura 5 – Formato típico de transmissão de dados serial</i>	34
<i>Figura 6 – Esquema de byte check sum</i>	36
<i>Figura 7 – Forma de onda correta</i>	38
<i>Figura 8 – Foto de PIC</i>	40
<i>Figura 9 – Esquema de funcionamento do MAX232</i>	42
<i>Figura 10 – Oscilador de cristal</i>	43
<i>Figura 11 – Forma de onda de oscilador</i>	43
<i>Figura 12 – Interferência de inicio de oscilação</i>	44
<i>Figura 13 – Foto módulo RF</i>	45
<i>Figura 14 – Esquema geral de funcionamento</i>	53
<i>Figura 15 – Esquema de ligação do emissor</i>	56
<i>Figura 16 – Esquema de ligação do receptor</i>	57
<i>Figura 17 – Ligações do PIC</i>	61
<i>Figura 18 - Efeito Doppler</i>	62
<i>Figura 19 – Diagrama de blocos do receptor RF</i>	63
<i>Figura 20 – Ligação do receptor RF</i>	64
<i>Figura 21 – Esquema elétrico do emissor</i>	64
<i>Figura 22 – Ligações do emissor RF</i>	65
<i>Figura 23 – Ligações do oscilador</i>	66
<i>Figura 24 – Ligações do regulador de tensão</i>	67
<i>Figura 25 – Esquema de ligação do RS232.</i>	68
<i>Figura 26 – Protocolo de Transmissão</i>	69
<i>Figura 27 – Via de duas mão 1</i>	72
<i>Figura 28 – Via de duas mãos 2</i>	73
<i>Figura 29 – Tela do terminal de comunicação serial</i>	74
<i>Figura 30 – Tela do gravador de PIC</i>	76
<i>Figura 31 – Esquema de ligação para testar as portas seriais</i>	77
<i>Figura 32 – Banco de dados das linhas de ônibus</i>	79

<i>Figura 33 – Teste do programa</i>	80
<i>Figura 34 – Diagrama de blocos do programa do PIC</i>	83
<i>Figura 35 – Esquema do programa do PIC</i>	84
<i>Figura 36 – Diagrama do sistema de integração</i>	88
<i>Figura 37 – Esquema do software leitor de porta serial</i>	89
<i>Figura 38 – Diagrama de blocos do leitor de porta serial</i>	90
<i>Figura 39 – Esquema do banco de dados</i>	101
<i>Figura 40 – Foto do emissor</i>	104
<i>Figura 41 – Foto do receptor</i>	105

LISTA DE TABELAS

<i>Tabela 1 - Faixas de Freqüência de RFID</i>	28
<i>Tabela 2 - Leis X Fórmulas</i>	29
<i>Tabela 3 - Vantagens RF X RFID</i>	48
<i>Tabela 4 - Desvantagens RF X RFID</i>	48
<i>Tabela 5 - Estimativa dos Custos de Implementação</i>	49
<i>Tabela 6 - Tabela de faixas de freqüências com restrição de uso</i>	50
<i>Tabela 7 – Tabela de limites gerais de emissão.</i>	50
<i>Tabela 8 - Constantes e valores da API do Windows (TM) utilizadas para configurar a Porta Serial</i>	94

LISTA DE CODIGOS FONTE

Código 1 – Código fonte que define o protocolo de comunicação	70
Código 2 – Código fonte que verifica o protocolo de comunicação	70
Código 3 – Código fonte para piscar LED	76
Código 4 – Código que testa a porta serial	78
Código 5 – Código que testa os botões	78
Código 6 – Monta um ID de ônibus.	84
Código 7 – Função que verifica a o botão apertado.	85
Código 8 – Função que cria e consolida a string de protocolos	86
Código 9 – Método de envio da porta serial	86
Código 10 - Uso da função CreateFile() da API do Windows	92
Código 11 - Uso das funções GetCommState() e SetCommState()	93
Código 12 - Uso das funções GetCommState(), SetCommState() e BuildCommDCB()	94
Código 13 - Uso da função GetCommTimeouts() e SetCommTimeouts()	95
Código 14 – Uso da função ReadFile()	95
Código 15 – Uso da função WriteFile()	96
Código 16 – Uso da função CloseHandle()	97
Código 17 – Código fonte para acesso ao banco de dados	102
Código 18 – Código fonte para tocar o som	103

SUMÁRIO

LISTA DE TABELAS	10
LISTA DE CODIGOS FONTE	11
LISTA DE CODIGOS FONTE	11
1 <i>Introdução</i>	15
2 <i>Objetivo</i>	17
3 <i>Motivação</i>	18
4 <i>Fundamentação Teórica</i>	20
4.1 <i>Sistemas RF</i>	22
4.1.1 Aplicações Mais Freqüentes	22
4.1.2 Sistemas RFID	23
4.1.3 Comparação da Comunicação via RF e RFID	23
4.1.4 Componentes do Sistema RFID	25
4.1.5 Classificação dos Sistemas de RFID	26
4.1.6 Aplicações do RFID	26
4.1.7 Faixa de Freqüência Utilizada	27
4.2 <i>Eletromagnetismo</i>	28
4.2.1 Teoria de Eletromagnetismo	29
4.2.2 Antenas	30
4.2.3 Efeito Doppler	31
4.3 <i>Comunicação Serial</i>	32
4.3.1 Taxa de Transferência (Baud Rate)	32
4.3.2 Sistemas Assíncronos	33
4.3.3 Checksum e Paridade	35
4.3.4 Interface Serial RS232	36
4.3.5 Temporizarão de Sinais	37
4.3.6 Conversores de nível TTL – RS232	38
4.4 <i>PIC</i>	39
4.5 <i>Banco de Dados</i>	41
4.6 <i>Hardware</i>	41
4.6.1 Max232	41
4.6.2 Osciladores	42
4.6.3 Modulo RF	44

5	Metodologia	46
5.1	Análise	46
5.1.1	Estudo de viabilidade	46
5.1.2	Vantagens e Desvantagens	48
5.1.3	Custos Comparativos	48
5.1.4	Legislação e Homologação	50
5.2	Projeção	51
5.3	Construção	51
6	Projeto	52
6.1	Introdução	52
6.2	Alterações no Projeto	53
6.3	Características Técnicas	54
6.3.1	Emissor	55
6.3.2	Receptor	55
6.4	Esquema Completo de Funcionamento	56
6.5	Custo do Protótipo	57
6.5.1	Dispositivo móvel (emissor)	57
6.5.2	Receptor	58
6.6	Componentes Utilizados	58
6.6.1	Micro controlador PIC	58
6.6.2	Modulo de Radiofrequência	61
6.6.3	Osciladores	65
6.6.4	Reguladores de Tensão	66
6.6.5	Ligação Serial	67
6.7	Protocolo de Transmissão	68
6.8	Sistema de Logística	71
6.8.1	Detecção de Sinais repetidos	71
6.8.2	Vias de duas mãos	71
6.9	Ensaios Realizados	73
6.9.1	Ensaios com Regulador	73
6.9.2	Ensaios com a porta serial	74
6.9.3	Ensaios com o PIC	75
6.9.4	Ensaios com o Módulo RF	78
6.9.5	Ensaios com o Banco de Dados	79

6.9.6	Ensaios com o Tocador de Som	80
6.10	Programação do PIC	80
6.10.1	Software de Programação	81
6.10.2	Etapas para o desenvolvimento	81
6.10.3	Descrição de Funcionamento	82
6.11	Programação do Micro Computador	87
6.11.1	Esquema de integração	87
6.11.2	Leitor da porta serial	88
6.11.3	Comunicação com banco de dados	97
6.11.4	Tocar SOM	102
7	Resultados	103
8	Conclusão	107
9	Referência Bibliográfica	109

1 Introdução

Este trabalho apresenta o estudo e o projeto de construção do protótipo de um sistema de identificação de linhas de ônibus para o auxílio de deficientes visuais. De acordo com as atividades elaboradas nele demonstramos as etapas que o desenvolvimento do trabalho obteve para alcançarmos o devido cumprimento de nossa meta; a construção e elaboração de um trabalho acadêmico com uma função social prática.

Sugerido pela Comissão de Projetos de Graduação com Responsabilidade Social, composta pela Poli Cidadã, o tema deste trabalho deve contornar problemas que o uso do transporte público impõe aos seus usuários. Aqui apresentamos o esquema necessário para atender os requisitos indicados e avaliados ao decorrer da abordagem do tema a fim de solucionar o problema apresentado. Assim, empregando tecnologia associada com a análise dos aspectos relevantes aos requisitos podemos promover a melhoria da qualidade do transporte público, escopo de nosso projeto.

Levando em conta os aspectos limitantes quanto à formulação de uma solução, apresentamos a modelagem de um sistema de Radio Freqüência para a identificação dos veículos de transporte coletivo. As condições contornadas aqui vão desde aspectos econômicos até elementos físicos, que delimitam a estrutura do sistema proposto.

Para atender as condições avaliadas modelamos o sistema através de uma metodologia seguida de procedimentos para a definição de uma solução adequada. Assim, este foi desenvolvido através de módulos de Radio Freqüência manipulados por micro controladores programados e associados por meio de um software com um banco de dados contendo informações da linhas, possibilitando o monitoramento do tráfego dos ônibus através dos pontos do perímetro urbano das cidades. Desta forma, poderemos atender as especificações sugeridas de acordo com critérios estabelecidos, se baseado na análise de fatores relevantes ao desenvolvimento do protótipo.

Esse documento descreve o produto do trabalho realizado para a constituição do sistema de identificação de linha de ônibus.

2 Objetivo

Melhorar a vida do usuário do transporte coletivo e das operadoras de ônibus é a idéia por trás desse projeto. Mais especificamente este trabalho visa, através da elaboração e do desenvolvimento de um sistema de identificação de linhas de ônibus, construir um protótipo que pode auxiliar deficientes visuais. Assim, o objetivo é constituir um sistema capaz de efetuar tal tarefa associado à construção de um protótipo do mesmo.

Este protótipo baseia-se em módulos Radio Freqüência de transferência de dados entre transmissor e receptor e tem a função de realizar a comunicação remota entre dois aparelhos eletrônicos. Estes instalados nos ônibus e em antenas localizadas nos pontos de ônibus possibilitam a constituição do sistema de identificação destas linhas de ônibus, que podem ser então anunciadas por alto-falantes.

A estrutura, assim, possibilita que seja montado na rede de transporte público um mecanismo de monitoramento das linhas de ônibus através do registro dos veículos que trafegam pelas regiões equipadas pelo sistema. Através da comunicação entre os terminais RF montados, o posicionamento dos veículos pode ser determinado. Desta maneira, as informações procedentes podem ser também pós-processadas e avaliadas a fim de solucionar problemas existentes no tráfego e no abastecimento da rede viária em que se tem o mecanismo.

3 Motivação

A escolha deste tema para trabalho de formatura tem como motivação a melhoria da qualidade do serviço prestado pelo transporte público a todos os seus clientes, deficiente ou não. Isto, visto que atualmente pontos de ônibus em muitas ocasiões ficam cheios, resultando aos usuários dificuldades em identificar a linha de ônibus do veículo que se aproxima, pois os letreiros não são grandes suficientes e as pessoas nem sempre estão prestando a devida atenção.

Instalando antenas pela cidade para rastrear os ônibus pode-se começar o rastreamento destes automóveis. Horários de atendimento, velocidade do percurso, identificação de lentidão e etc. Com isso também será possível o levantamento do perfil do trânsito da cidade, disponibilizando informações para um estudo na redução dos congestionamentos.

Assim, através deste aprimoramento podemos possibilitar a melhoria da qualidade de vida associada a esse serviço imprescindível que é o transporte público. Isso tanto na esfera ambiental, com o possível aumento da utilização de ônibus ao invés do carro, quanto na esfera de social, pelo incentivo de uso e pela sua inclusão igualitária.

Outro motivo pelo qual este projeto gera motivação é que a tecnologia de RF vem se desenvolvendo rapidamente nos últimos tempos. Uma das técnicas que se destaca é o RFID (Radio Frequency Identification), que tem ampliado sua atuação e que, pela análise, geral podemos constatar que será muito explorada em novas aplicações.

A tecnologia de Radiofreqüência estará capacitada a superar os conceitos de atendimento e agilidade no varejo e controle, e a aperfeiçoar a rastreabilidade dos produtos em toda a cadeia de suprimentos. Muito se tem falado sobre suas perspectivas e as consequências advindas para os usuários que hoje trabalham com o código de barras. Como tais tecnologias conviverão lado a lado? Qual será o futuro com chips rastreadores?

Com a utilização de RF há uma série de benefícios: fazer a leitura de itens sem a proximidade do leitor, permitindo, por exemplo, a contagem instantânea de estoque, a verificação imediata dos produtos nas prateleiras

ou no "carrinho" do varejo; melhorar práticas de reabastecimento com eliminação de itens faltantes e itens com validade vencida; identificar a localização dos itens em processos de recall (busca) e outras possibilidades ilimitadas de melhoria e individualização de serviços ao consumidor. O fluxo físico de produtos e a comunicação entre numerosos parceiros de uma variedade e quantidade de itens tornam o processo cada vez mais custoso em toda a cadeia de suprimentos

Podemos ter uma idéia desse esforço de desenvolvimento pelo sistema EPC Global. Esse foi idealizado para funcionar como uma "Internet de Objetos". Cada item terá o seu próprio número individual codificado em uma etiqueta de radiofreqüência (RFID). Os leitores farão a captura desta identificação e serão capazes de localizar onde o item se encontra e as condições do mesmo, informando bancos de dados remotos através da Internet. Com isso, consegue-se a identificação automática e a rastreabilidade de produtos em tempo real. O objetivo é criar um padrão aberto e global para esta identificação e para a troca de informações à medida que os produtos percorrerem a cadeia de suprimentos.

A EPC Global é uma organização sem fins lucrativos, criada pela EAN International e a UCC (Uniform Code Council), numa parceria 50% / 50%, que terá sob sua responsabilidade aspectos de comercialização e concessão do uso do Sistema EPC. Tais atribuições envolvem, principalmente, o gerenciamento do banco de dados de produtos, as questões de propriedade intelectual, o desenvolvimento de padrões técnicos do Sistema EPC, o desenvolvimento da sua tecnologia, e ainda a comunicação, o suporte ao usuário e todas as fases de educação e treinamento. O contrato existente entre a UCC e o MIT é uma licença de uso das patentes sobre o sistema EPC. O propósito da EPC Global é dar um escopo global deste contrato por meio da EAN International e de suas organizações.

Desta maneira o envolvimento dessa tecnologia em nosso projeto nos atrai para o estudo e confecção do sistema proposto.

4 Fundamentação Teórica

Neste capítulo pode-se encontrar um estudo detalhado sobre equipamentos com funcionalidade similares ao nosso sistema, uma breve teoria sobre cálculos e dimensionamentos relevantes utilizados na elaboração do projeto e uma análise sobre os principais componentes utilizados no protótipo.

4.1 Aplicações Existentes

A fim de determinar a utilização de uma tecnologia barata e inovadora em termos de aplicação com esta funcionalidade foi feito um estudo sobre quais as tecnologias utilizadas ou em fase de estudos disponíveis no mundo na identificação de ônibus. Foram verificados dois projetos pioneiros e assim, mais difundidos no mundo, este capítulo tem como objetivo apresentar estes projetos e os locais em que esses estão sendo mais difundidos.

Um sistema GPS acoplado a um ônibus está sendo utilizado para garantir uma maior acurácia dos tempos de chegada de ônibus nos seus respectivos pontos, porém apenas GPS não é suficiente para atender passageiros de forma adequada quanto ao ETA (tempo estimado para chegada), assim diversos projetos pilotos estão sendo testados em varias cidades do mundo, utilizando RFID com GPS. Nos Estado Unidos existem alguns pontos de ônibus que tem um receptor RFID que transmite informações via ZigBee (Ref [ZIGBEE]) conexão sem fio quando o ônibus se aproxima do ponto. A vantagem deste dispositivo é indicar o momento exato em que o ônibus está se aproximando. Assim pode-se ter mais acurácia do tempo de chegada em cada ponto. Este sistema também pode ser encontrado em Hong Kong com a implementação em mais de 5.000 veículos.

Durante nossa pesquisa sobre dispositivos utilizados em outras cidades do mundo, identificamos um sistema inteligente de transporte

coletivo chamado *Bus rapid transit* Ref[BRT], Para as cidades que estão utilizando este sistema os ônibus são monitorados via GPS.

Entre outras utilidades deste sistema esta o monitoramento dos veículos de transporte. Cada ônibus recebe sua posição por GPS e envia os dados para uma antena de transmissão, esta por sua vez analisa o fluxo de transito do local e envia aos pontos de ônibus informações como tempo esperado para a chegada do ônibus.

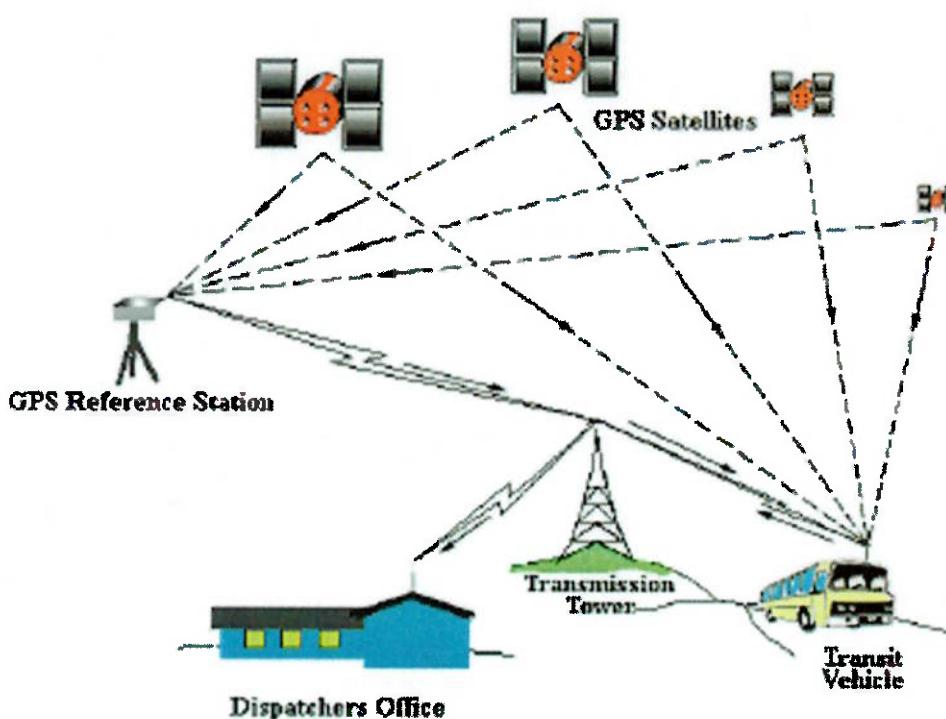


Figura 1 – Sistema de monitoramento por GPS

As principais cidades do mundo que já estão utilizando ou está sendo implementado este sistema são:

- América do Norte: Charlotte, Cleveland, Eugene, Hartford, Houston, Los Angeles, Miami-Dade, New York City, Pittsburgh, Seattle (Estados Unidos) , Ottawa (Canada).
- Oceania: Adelaide, Brisbane, Sydney (Austrália).
- Europa: Leeds, Rouen (França), Runcorn (UK).
- América do Sul: Belo Horizonte, Porto Alegre, São Paulo (Brasil), Bogotá (Colômbia), Quito (Equador)

4.2 Sistemas RF

Um sistema RF pode ser definido como toda forma de transmissão de dados ou mesmo uma única informação via rádio freqüência, ondas de rádio [Wikipedia]. Sendo assim existem inúmeros instrumentos que utilizam esta tecnologia de diversas formas.

Veja a seguir um breve resumo sobre as aplicações, alcance, freqüências mais utilizadas e o funcionamento básico de alguns sistemas que utilizam RF.

4.2.1 Aplicações Mais Freqüentes

Praticamente todos os dispositivos que possuem comunicação sem fio utilizam sistemas RF. Desde o controle para abrir o portão de uma garagem até a comunicação entre dois micros computadores via “wireless”. Falaremos nesta seção um pouco sobre sistemas RFs mais comuns, que utilizam sistema de identificação ou transferência de dados.

Um exemplo típico de identificação via RF é os controles de aberturas de portões automáticos ou acionamento de alarmes sem fio. Estes sistemas utilizam freqüências de 292MHz ou 433,92MHz com code learning, [RF Aplicações], isto uma forma de encriptar o sinal para que cada controle acione somente o sistema pré-determinado. Este sistema consiste um receptor e um transmissor de dados, com um *encoder* e um *decoder*. Assim uma combinação de Tensões (0 ou 5V) geram um número binário que é transmitido via paralelo para o *encoder*. O *encoder* envia os sinais em uma determinada taxa de transmissão de forma serial para o módulo RF. O sistema receptor recebe os sinais e decodifica, assim possibilitando comparar os sinais recebidos com um padrão do sistema para ver se este foi um código de acionamento de algum sistema. Este tipo de sistema tem um alcance médio de 40m.

Outra utilização muito comum de RF é as rádios AM e FM. Estes sistemas utilizam freqüência de AM vai de 540 a 1600 kHz e FM de 88 a 108MHz [ABTU]. A diferença fundamental entre este sistema RF e o citado anteriormente é o alcance alcançado por este. Isto se deve devido à faixa de freqüência, menor, e a potência de emissão dos dados. Para gerar esta potência na emissão dos dados utiliza-se um transmissor. A função do transmissor é transformar o sinal elétrico, proveniente da saída de áudio de um aparelho (como a saída para fone de ouvido de um *walkman*, por exemplo), em um sinal eletromagnético, que contém, de forma codificada, a informação do áudio original, este sinal eletromagnético é distribuído no espaço através da antena. O alcance depende muito da potência da fonte emissora e do relevo da mesma.

Outro exemplo que é interessante ser comentado é a comunicação de rede *wireless*. Este sistema utiliza freqüência de 2,4 GHz (802.11b) ou 5 GHz (802.11a) [wirelessbrasil]. As diferenças fundamentais entre este sistema e os comentados anteriormente é a freqüência utilizada que é muito maior e que é um sistema de malha fechada, isto é, todos os objetos do sistema recebem e enviam dados. O alcance médio é de 30 a 100 metros.

4.2.2 Sistemas RFID

O RFID é um caso particular de RF porem deve ser analisado de forma diferenciada, pois está fase em desenvolvimento e sua aplicação tem visto um crescimento acentuado atualmente. Veja a seguir um breve resumo sobre as aplicações e o funcionamento de alguns sistemas que utilizam RFID. Arquitetura e Tecnologia do RFID. [Wikipedia]

Antes que RFID possa ser compreendido completamente, é essencial compreender como uma comunicação por radio freqüência ocorre.

4.2.3 Comparação da Comunicação via RF e RFID

Uma comunicação via RF (freqüência de rádio) ocorre pela transferência de dados através de ondas eletromagnéticas. Gerando uma onda eletromagnética com uma freqüência específica na fonte, seu efeito pode ser observado no receptor longe da fonte, que o identifica assim e então a coleta informação. Essa identificação se dá pela indução de uma corrente pela passagem de um campo magnético por um solenóide que, ligado a um chip, transmite informações gravadas. [UFG]

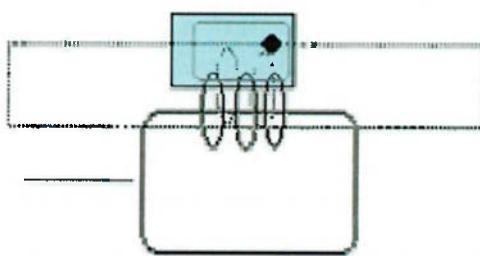


Figura 2 - Indução Magnética do TAG pela Antena

Em um sistema de RFID, o *Tag* de RFID que contém os dados, etiquetados no objeto, gera um sinal que contém a informação respectiva que é lida pelo leitor de RFID, que então pode passar esta informação a um processador para interpretar a informação obtida para essa aplicação particular. Esta informação segue uma padronização quanto ao seu formato de identificação. Com dados referentes ao número de série, dado gravado e horário da aferição. [GTA]

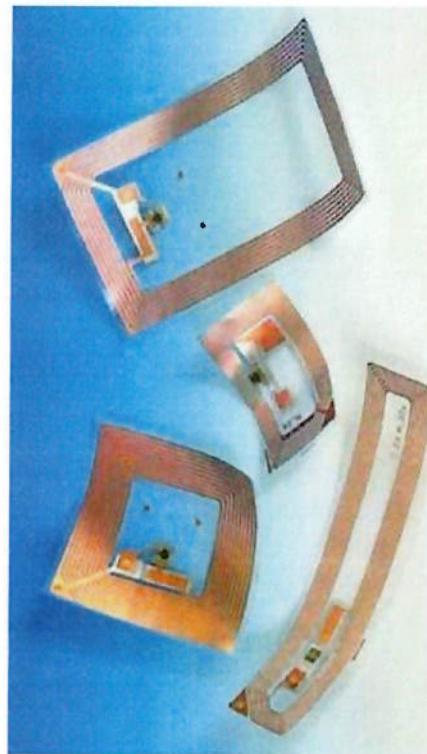


Figura 3 - Tag

4.2.4 Componentes do Sistema RFID

Assim, um sistema de RFID pode ser visualizado como a soma dos seguintes três componentes:

- Tag ou transponder de RFID
- Leitor ou transceptor de RFID
- Subsistema de processo de dados

Um *Tag* de RFID é composto de uma antena, de um transponder sem fio e de um material encapsulado. Estes *Tags* podem ser ativos ou passivos. Enquanto os *Tags* ativos têm necessidade de fonte externa de energia (uma bateria), os *Tags* passivos usam a energia induzida pelo campo magnético da antena de RFID. Assim os *Tags* passivos são mais baratos, mas com alcance (distância de leitura) muito menor (<<<10m) [Wikipedia] e mais sensível às condições de regulação e ambientais, em comparação aos *Tags* ativos.

Um leitor de RFID consiste em uma antena, um transceptor e um decodificador, que emite sinais periódicos que incidem sobre todos os Tag na vizinhança. Ao receber o sinal de um Tag passa essa informação ao processador de dados.

O subsistema de processamento de dados fornece os meios de interpretar e de armazenar os dados.

4.2.5 Classificação dos Sistemas de RFID

Os sistemas de RFID podem ser classificados conforme a escala de freqüência utilizada.

De 30 kHz a 500 kHz e 13,56 MHz, são os sistemas de baixa freqüência *low-frequency* têm escalas curtas da leitura e abaixam custos do sistema. São usados mais geralmente em sistemas de segurança e nos sistemas de identificação de animais.

De alta freqüência (850 MHz a 950 MHz; 2.4 GHz a 2.5 GHz e 5.8GHz), possibilidade de leituras a longa distância (até de 27m) e velocidades da leitura elevada, são utilizadas em aplicações como cobrança automatizada do pedágio (no nosso caso, leitura de linha de ônibus) [Wikipedia].

Apesar do desempenho mais elevado de sistemas de alta freqüência de RFID os custos são mais elevados que sistemas de baixa freqüência.

4.2.6 Aplicações do RFID

Há duas principais áreas de aplicação: uma amplamente definida como de proximidade (escala curta) e outra definida como vizinhança (escala longa).

As aplicações da escala longa ou da vizinhança podem geralmente ser descritas como aplicações de trilha e de percurso, mas a tecnologia fornece a funcionalidade e benefícios adicionais para a autenticação do produto.

O RFID permite uma automatização maior do processo de levantamento de dados. A maioria das companhias gasta um esforço considerável em saber o que está em seu armazém. O RFID ajudar-lhes-á escavar muito mais facilmente e mais profundamente as informações de cada produto, seguindo em detalhe os padrões de qualidade de cada unidade, logo depois que saiu da fábrica ou do armazém.

O RFID também permite que todos os dados sejam transferidos com alta confiabilidade. As companhias usam fornecedores independentes, os dados de cada um deles podem ser gravados nos “Tags” e lidos no sistema central da companhia.

Desta maneira é possível o controle da companhia no ciclo de vida de um produto. Poderá ser mais bem entendido as falhas e os sucessos. Numerosos casos foram registrados nos quais companhias tiveram que recolher todos os produtos devido a uma falha em um componente menor. Os custos envolvidos em recolher um carro inteiro por um erro no sistema de câmbio, por exemplo. Haveria uns dados melhores sobre o desempenho da produção do borne. Um carro poderia individualmente ser etiquetado em seus componentes. Os dados podiam ser coletados em toda parte, locais do acidente, lojas de reparo, mesmo nas garagens. Analisando dentro da fábrica, a “Tag” poderia permitir um levantamento focalizado na falha mais rapidamente.

Algumas outras áreas onde RFID passivo são aplicados atualmente são:

- 1 Identificação de Pessoa
- 2 Controle de Produção do Alimento
- 3 Monitoração do Estacionamento de Veículo
- 4 Gestão de Recurso
- 5 Controle de Acesso

4.2.7 Faixa de Freqüência Utilizada

Existe uma diversidade muito grande de aparelhos que operam com radio freqüência. Alguns exemplos das respectivas faixas ocupadas por estes são mostrados abaixo.

Veja abaixo as faixas de freqüência de RFID determinada para cada tipo de instrumento [UFG].

Tabela 1 - Faixas de Freqüência de RFID

Faixa de Freqüência	Características	Aplicações típicas
Baixa 100-500 kHz	Curto a médio raio de leitura Baixo custo Baixa velocidade de leitura	Controle de acesso em geral Identificação de animais Depósitos de almoxarifados Garagens de veículos
Intermediária 10-15 MHz	Curto a médio raio de leitura Custo considerável Média velocidade de leitura	Controle de acesso especial Smart cards
Alta 850-950 MHz 2.4-5.8 GHz	Grande raio de alcance Leitura em alta velocidade Linha de vista necessária Caro	Monitoramento de veículo em trânsito Aplicações mais sensíveis

4.3 Eletromagnetismo

Visto que a radiofreqüência é uma onda eletromagnética, neste item procuramos fazer uma breve apresentação sobre o tema. A radiação eletromagnética é uma combinação de um campo elétrico e de um campo magnético que se propagam através do espaço transportando energia. Dependendo das circunstâncias, esta radiação pode comportar-se como uma onda ou como uma partícula.

Tanto a luz como o infravermelho ou ondas de rádios, são ondas eletromagnéticas, o que as diferencia é a freqüência. Quanto mais alta for

essa freqüência mais energética é a onda. As ondas eletromagnéticas têm como característica principal a sua velocidade. Considerada a maior velocidade do universo, elas podem vencer vários obstáculos físicos, tais como gases, atmosfera, água, paredes, dependendo da sua freqüência. As ondas eletromagnéticas viajam com velocidade constante no vácuo a cerca de 300.000 km/s (velocidade da luz).

4.3.1 Teoria de Eletromagnetismo

No estudo da Física, Michael Faraday e James Clark Maxwell foram os precursores desta área. Eles chegaram a conclusões utilizando cálculos diferenciais e integrais para explicar a relação entre a eletricidade e o magnetismo.

Quando o campo eletromagnético é estacionário não há propagação de informação através do espaço. Quando o campo eletromagnético é variável, há propagação das modificações desde a fonte do campo magnético através do espaço sob a forma de uma onda. São exemplos de campos eletromagnéticos variáveis as ondas de rádio, as microondas, a luz, os raios X, e os raios gama.

A base teórica do eletromagnetismo é o conjunto das quatro equações de Maxwell, que descrevem como os campos elétrico e magnético se relacionam e como variam em função do tempo e da posição no espaço. Estas equações constituem mesmo o nascimento do eletromagnetismo, já que antes dos estudos de Maxwell os capítulos da eletricidade e do magnetismo eram tidos como aspectos separados da Física. [Halliday]

Tabela 2 - Leis X Fórmulas

Nome	Diferencial	Forma Integral
Lei de Gauss:	$\nabla \cdot D = \rho$	$\oint_s D \cdot ds = Q_{englobado}$
Lei de Gauss para o magnetismo (ausência de)	$\nabla \cdot B = 0$	$\oint B \cdot ds = 0$

monopolos magnéticos):		
Lei da indução de Faraday:	$\nabla \times E = -\frac{\partial B}{\partial t}$	$\oint_s E \cdot dl = -\int_{\partial C} \frac{dB}{dt} \cdot ds$
Lei de Ampère + extensão de Maxwell:	$\nabla \times H = J + \frac{\partial D}{\partial t}$	$\oint_C H \cdot dl = I_{englobado} + \frac{d\phi_D}{dt}$

Em que:

p é a intensidade de carga elétrica livre (unidade SI: Coulomb por metro cúbico), não incluindo dipólos de cargas ligadas no material

B é a intensidade de fluxo magnético (unidade SI: tesla), também chamada de indução magnética.

D é o campo elétrico de deslocamento (unidade SI: coulomb por metro quadrado).

E é o campo elétrico (unidade SI: volt por metro),

H é o campo magnético (unidade SI: ampère por metro)

J é a densidade de corrente elétrica (unidade SI: ampère por metro quadrado)

4.3.2 Antenas

Um aspecto importante do eletromagnetismo em diversas aplicações é que quando a radiação eletromagnética atravessa um condutor elétrico induz uma corrente elétrica no condutor. Este efeito é utilizado nas antenas, amplamente empregadas atualmente.

Antena é o dispositivo cuja função é transformar energia eletromagnética guiada pela linha de transmissão em energia eletromagnética irradiada, pode-se também dizer que esta lei serve também no sentido inverso, isto é, transformar energia eletromagnética irradiada em energia eletromagnética guiada para a linha de transmissão. Portanto, sua função é primordial em qualquer comunicação onde exista radiofreqüência.

A relação entre as potências de emissão e recepção é proporcional e obedece à Fórmula de Friis.

A Fórmula de Friis (ou Lei de Friis) é usada em engenharia de telecomunicações e relaciona a potência transmitida de uma antena para outra em determinadas condições ideais.

Na sua forma mais simples, a Fórmula de Friis é a que se segue. Dadas duas antenas, o fator da potência recebida pela antena de recepção, P_r , sobre a potência transmitida à antena de transmissão, P_t , é dado por: [Wikipedia]

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi R} \right)^2$$

onde G_t e G_r são os ganhos das antenas de transmissão e recepção, respectivamente, λ é o comprimento de onda e R é a distância entre elas. Os ganhos das antenas são medidos com respeito a antenas isotrópicas (em unidades lineares e não em decibéis), com o comprimento de onda e a distância nas mesmas unidades.

4.3.3 Efeito Doppler

Tendo em vista que nosso projeto envolve a emissão e captação de ondas de radio em objetos com velocidade devemos considerar os efeitos desta sobre o fenômeno de transmissão.

Em 1842 Christian Doppler (1803-1853) deduziu que para um corpo luminoso se aproximando (ou se afastando) do observador o comprimento de onda da luz diminui (aumenta), em relação àquele observado em laboratório. O comprimento de onda de uma fonte que está se movimentando com velocidade v em relação ao observador é deslocado por: [Wikipedia]

$$\frac{\Delta\lambda}{\lambda} = \frac{v}{c} \left(\frac{1}{1 - \frac{v^2}{c^2}} \right)$$

onde θ é o ângulo entre o vetor velocidade e a linha de visada. Se a velocidade for muito menor que a velocidade da luz, e considerando v_r como a componente de velocidade na direção do observador: [Wikipedia]

$$\frac{\Delta\lambda}{\lambda} = \frac{v_r}{c}$$

4.4 Comunicação Serial

A maioria das mensagens digitais são mais longas que alguns poucos bits. Por não ser prático nem econômico transferir todos os bits de uma mensagem simultaneamente, a mensagem é quebrada em partes menores e transmitida seqüencialmente. A transmissão bit-serial converte a mensagem em um bit por vez através de um canal. Cada bit representa uma parte da mensagem. Os bits individuais são então rearranjados no destino para compor a mensagem original. Em geral, um canal irá passar apenas um bit por vez. A transmissão bit-serial é normalmente chamada de transmissão serial, e é o método de comunicação escolhido por diversos periféricos de computadores.

A transmissão byte-serial converte 8 bits por vez através de 8 canais paralelos. Embora a taxa de transferência seja 8 vezes mais rápida que na transmissão bit-serial, são necessários 8 canais, e o custo poderá ser maior do que 8 vezes para transmitir a mensagem. Quando as distâncias são curtas, é factível e econômico usar canais paralelos como justificativa para as altas taxas de transmissão. A interface Centronics de impressoras é um caso típico de transmissão byte-serial. [MINICURSO]

4.4.1 Taxa de Transferência (Baud Rate)

A taxa de transferência refere-se a velocidade com que os dados são enviados através de um canal e é medido em transições elétricas por segundo. Na norma EIA232, ocorre uma transição de sinal por bit, e a taxa de transferência e a taxa de bit (bit rate) são idênticas. Nesse caso, uma taxa de 9600 bauds corresponde a uma transferência de 9600 dados por segundo, ou um período de aproximadamente, 104 ms (1/9600 s).

Outro conceito é a eficiência do canal de comunicação que é definido como o número de bits de informação utilizável (dados) enviados através do canal por segundo. Ele não inclui bits de sincronismo, formatação, e detecção de erro que podem ser adicionados a informação antes da mensagem ser transmitida, e sempre será no máximo igual a um.
 [MINICURSO]

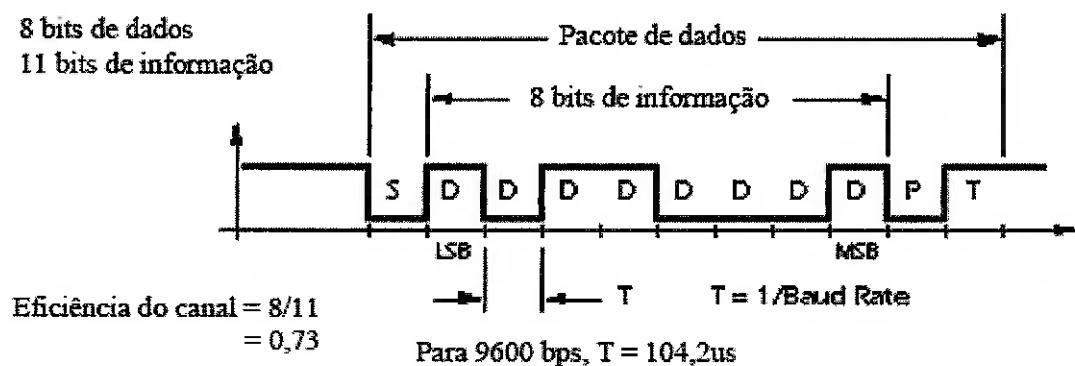


Figura 4 – Ondas para transmissão serial

4.4.2 Sistemas Assíncronos

Em sistemas assíncronos, a informação trafega por um canal único. O transmissor e o receptor devem ser configurados antecipadamente para que a comunicação se estabeleça a contento. Um oscilador preciso no receptor irá gerar um sinal de clock interno que é igual (ou muito próximo) ao do transmissor. Para o protocolo serial mais comum, os dados são enviados em pequenos pacotes de 10 ou 11 bits, dos quais 8 constituem a mensagem. Quando o canal está em repouso, o sinal correspondente no canal tem um nível lógico '1'. Um pacote de dados sempre começa com um nível lógico '0'

(start bit) para sinalizar ao receptor que um transmissão foi iniciada. O “start bit” inicializa um temporizador interno no receptor avisando que a transmissão começou e que serão necessários pulsos de clocks. Seguido do start bit, 8 bits de dados de mensagem são enviados na taxa de transmissão especificada. O pacote é concluído com os bits de paridade e de parada (“stop bit”).

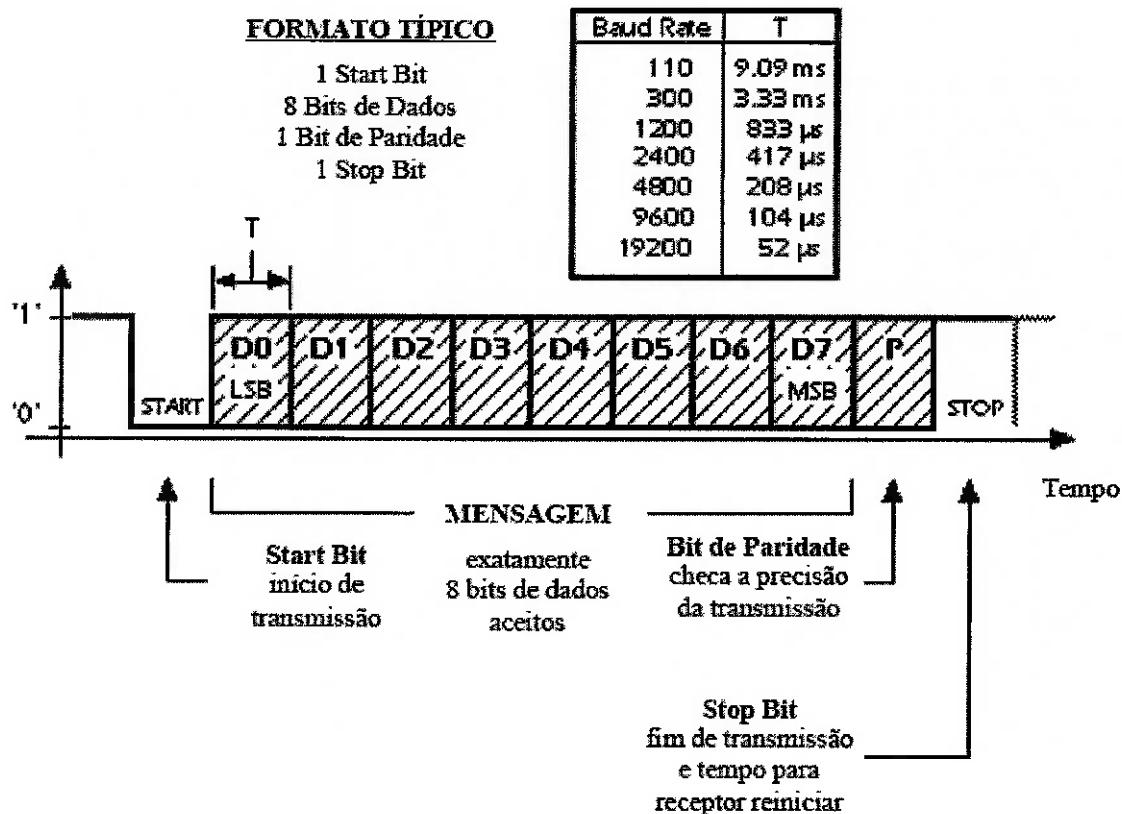


Figura 5 – Formato típico de transmissão de dados serial

O comprimento do pacote de dados é pequeno em sistemas assíncronos para minimizar o risco do oscilador do transmissor e do receptor variar. Quando osciladores a cristal são utilizados, a sincronização pode ser garantida sobre os 11 bits de período. A cada novo pacote enviado, o *start bit* “reseta” a sincronização, portanto a pausa entre pacotes pode ser longa. [MINICURSO]

4.4.3 Checksum e Paridade

Ruídos e distúrbios elétricos momentâneos podem causar mudanças nos dados quando estão trafegando pelos canais de comunicação. Se o receptor falhar ao detectar isso, a mensagem recebida será incorreta, resultando em consequências possivelmente sérias. Como uma primeira linha de defesa contra erros de dados, eles devem ser detectados. Se um erro pode ser sinalizado, pode ser possível pedir que o pacote com erro seja reenviado, ou no mínimo prevenir que os dados sejam tomados como corretos. Se uma redundância na informação for enviada, 1 ou 2 bits de erros podem ser corrigidos pelo hardware no receptor antes que o dado chegue ao seu destino.

O bit de paridade é adicionado ao pacote de dados com o propósito de detecção de erro. Na convenção de paridade-par (even-parity), o valor do bit de paridade é escolhido de tal forma que o número total de dígitos '1' dos dados adicionado ao bit de paridade do pacote seja sempre um número par. Na recepção do pacote, a paridade do dado precisa ser recomputada pelo hardware local e comparada com o bit de paridade recebido com os dados. Se qualquer bit mudar de estado, a paridade não irá coincidir, e um erro será detectado. Se um número para de bits for trocado, a paridade coincidirá e o dado com erro será validado. Contudo, uma análise estatística dos erros de comunicação de dados tem mostrado que um erro com bit simples é muito mais provável que erros em múltiplos bits na presença de ruído randômico. Portanto, a paridade é um método confiável de detecção de erro.

Dado Bit de Paridade

1 0 1 1 0 0 1 0 0

1 0 0 0 1 0 1 0 1

Outro método de detecção de erro envolve o cálculo de um *checksum* quando mensagens com mais de um byte são transmitidas pelo canal de comunicação. Nesse caso, os pacotes que constituem uma mensagem são

adicionados aritmeticamente. Um número de *checksum* é adicionado a seqüência do pacote de dados de tal forma que a soma dos dados mais o *checksum* é zero.

Quando recebido, os dados devem ser adicionados pelo processador local. Se a soma do pacote der resultado diferente de zero, ocorreu um erro. Na ocorrência de erros é improvável (mas não impossível) que qualquer corrupção de dados resultem em *checksum* igual a zero.

$$\begin{array}{r} 10110001 \\ 10000110 \\ + 01001100 \text{ Dados} \\ \hline 11111111 \\ 10100000 \\ \hline 001100100010 \text{ Soma Aritmética} \end{array}$$
$$\begin{array}{r} 00100010 \text{ Soma truncada - 8 bits} \\ + 11011110 \text{ Checksum (complemento de 2)} \\ \hline 00000000 \text{ Soma + Checksum = 0} \end{array}$$

Figura 6 – Esquema de byte check sum

Podem ocorrer erros que não sejam apenas detectados, mas também sejam corrigidos se código adicional for adicionado a seqüência de dados do pacote. A correção de erros em uma transmissão, contudo, abaixa a eficiência do canal, e o resultado é uma queda na transmissão.
[MINICURSO]

4.4.4 Interface Serial RS232

RS é uma abreviação de *Recommended Standard*. Ela relata uma padronização de uma interface comum para comunicação de dados entre equipamentos, criada no início dos anos 60, por um comitê conhecido atualmente como *Electronic Industries Association* (EIA). Naquele tempo, a comunicação de dados compreendia a troca de dados digitais entre um computador central (*mainframe*) e terminais de computador remotos, ou

entre dois terminais sem o envolvimento do computador. Estes dispositivos poderiam ser conectados através de linha telefônica, e consequentemente necessitavam um modem em cada lado para fazer a decodificação dos sinais. Dessas idéias nasceu o padrão RS232. Ele especifica as tensões, temporizações e funções dos sinais, um protocolo para troca de informações, e as conexões mecânicas.

A mais de 30 anos desde que essa padronização foi desenvolvida, a EIA publicou três modificações. A mais recente, EIA232E, foi introduzida em 1991. Ao lado da mudança de nome de RS232 para EIA232, algumas linhas de sinais foram renomeadas e várias linhas novas foram definidas.

Embora tenha sofrido poucas alterações, muitos fabricantes adotaram diversas soluções mais simplificadas que tornaram impossível a simplificação da padronização proposta. As maiores dificuldades encontradas pelos usuários na utilização da interface RS232 incluem pelo menos um dos seguintes fatores:

- A ausência ou conexão errada de sinais de controle, resultam em estouro do buffer (*overflow*) ou travamento da comunicação.
- Função incorreta de comunicação para o cabo em uso, resultam em inversão das linhas de Transmissão e Recepção, bem como a inversão de uma ou mais linhas de controle (*handshaking*).

Felizmente, os *drivers* utilizados são bastante tolerantes aos abusos cometidos, e os *Clis* normalmente sobrevivem. [MINICURSO]

4.4.5 Temporizarão de Sinais

A norma EIA232 especifica uma taxa máxima de transferência de dados de 20.000 bits por segundo (o limite usual é 19200 bps). *Baud rates* fixos não são fornecidos pela norma. Contudo, os valores comumente usados são 300, 1200, 2400, 4800, 9600 e 19200 bps.

Mudanças no estado dos sinais de nível lógico “1” para “0” ou vice-versa devem seguir diversas características, dadas a seguir:

- Sinais que entram na zona de transição durante uma mudança de estado deve atravessar essa região com direção ao estado oposto sem reverter a direção ou reentrar;
- Para os sinais de controle, o tempo na zona de transição deve ser menor do que 1ms;

Para sinais de temporização, o tempo para atravessar a zona de transição deve ser:

- Menor do que 1 ms para períodos de bits maiores que 25 ms;
- 4% do período de um bit para períodos entre 25 ms e 125 ms;
- Menor do que 5 ms para períodos menores que 125 ms.

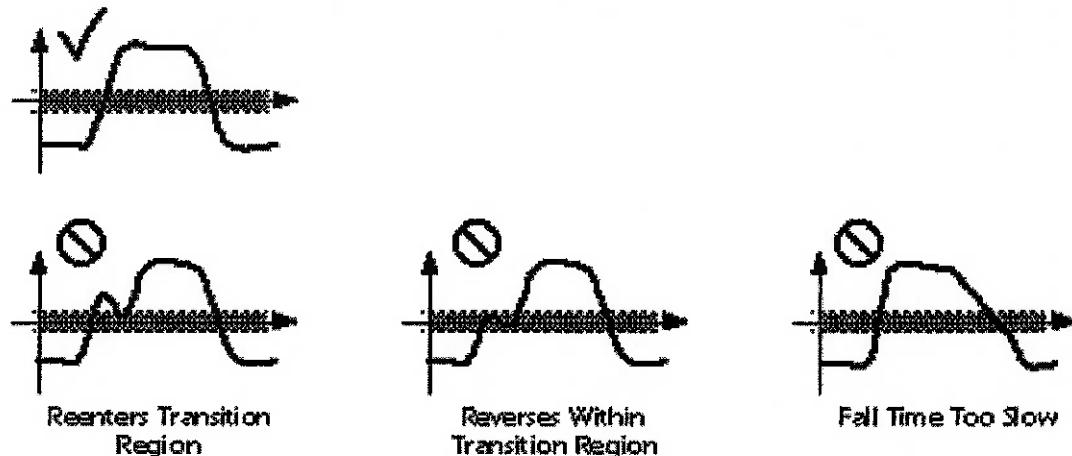


Figura 7 – Forma de onda correta

As rampas de subida e de descida de uma transição não devem exceder 30 V/ms. Taxas maiores do que esta, podem induzir sinais em condutores adjacentes de um cabo. [MINICURSO]

4.4.6 Conversores de nível TTL – RS232

A maioria dos equipamentos digitais utilizam níveis TTL ou CMOS. Portanto, o primeiro passo para conectar um equipamento digital a uma interface RS232 é transformar níveis TTL (0 a 5 volts) em RS232 e vice-versa. Isto é feito por conversores de nível.

Existe uma variedade grande de equipamentos digitais que utilizam o *driver* 1488 (TTL => RS232) e o *receiver* 1489 (RS232 => TTL). Estes CIs contém 4 inversores de um mesmo tipo, sejam *drivers* ou *receivers*. O *driver* necessita duas fontes de alimentação +7,5 volts a +15 volts e -7,5 volts a -15 volts. Isto é um problema onde somente uma fonte de +5 volts é utilizada.

Um outro CI que está sendo largamente utilizado é o MAX232 (da Maxim). Ele inclui um circuito de “charge pump” capaz de gerar tensões de +10 volts e -10 volts a partir de uma fonte de alimentação simples de +5 volts, bastando para isso alguns capacitores externos, conforme pode-se observar na figura a seguir. Este CI também tem 2 *receivers* e 2 *drivers* no mesmo “encapsulamento”.

Nos casos onde serão implementados somente as linhas de transmissão e de recepção de dados, não seria necessário 2 chips e fontes de alimentação extras. [MINICURSO]

4.5 PIC

O PIC é um circuito integrado produzido pela Microchip Technology Inc., que pertence à categoria dos microcontroladores, ou seja um componente integrado que num único dispositivo contém todos os circuitos necessários para realizar um sistema digital programável completo.

O PIC (neste caso um PIC16F877) pode ser visto externamente como um circuito integrado TTL ou CMOS normal, mas internamente dispõe de todos os dispositivos típicos de um sistema microprocessado, ou seja:

- Uma CPU (*Central Processor Unit* ou seja Unidade de Processamento Central) cuja finalidade é interpretar as instruções de programa
- Uma memória EPROM (*Erasable Programmable Read Only Memory* ou Memória Programável Somente de Leitura) na qual irá memorizar de maneira permanente as instruções do programa
- Uma memória RAM (*Random Access Memory* ou Memória de Acesso Aleatório) utilizada para memorizar as variáveis utilizadas pelo programa

- Um conjunto de LINHAS de I/O para controlar dispositivos externos ou receber impulsos de sensores, interruptores, etc.
- Um conjunto de dispositivos auxiliares ao funcionamento, ou seja gerador de *clock*, bus, contador, etc.
- A presença de todos estes dispositivos num espaço extremamente pequeno, dá ao projetista uma enorme gama de possibilidades e enormes vantagens em usar um sistema microprocessado onde, em pouco tempo e com poucos componentes externos, pode fazer o que seria oneroso com circuitos tradicionais.
- O PIC está disponível numa ampla gama de modelos para melhor se adaptar às exigências de projetos específicos, diferenciando-se pelo número de linhas de I/O e pelo conteúdo do dispositivo.
- Inicia-se com o modelo pequeno identificado pela sigla PIC12Cxx dotado de 8 pinos, indo até modelos maiores com sigla PIC18Cxxx dotados de 40 e/ou 80 pinos



Figura 8 – Foto de PIC

- Uma descrição detalhada da tipologia dos PIC está disponível no site da Microchip, acessível via internet, onde conseguimos encontrar grandes e variadas quantidades de informações técnicas, software de apoio, exemplos de aplicações e atualizações disponíveis.
- Para este projeto utilizamos um modelo de PIC o PIC16F877. Este é dotado de 40 pinos, sendo 31 disponíveis para I/O, ou seja, para serem ligados aos nossos circuitos e de algumas características que o tornam um circuito que melhor se adapta às exigências da nossa formação. A vantagem deste PIC é a existência de uma interface serial nas portas RC7 e RC6 pinos 25 e 26 respectivamente.
- Em particular o PIC16F877 dispõe de uma memória para armazenar o programa, do tipo FLASH, que pode ser reescrita quantas

vezes quisermos e que é ideal para as nossas experiências de programação. [ESTV]

4.6 Banco de Dados

Bancos de dados são conjuntos de dados com uma estrutura regular que organizam informação. Um banco de dados normalmente agrupa informações utilizadas para um mesmo fim. O modelo utilizado em nosso projeto é o modelo relacional, onde as estruturas têm a forma de tabelas.

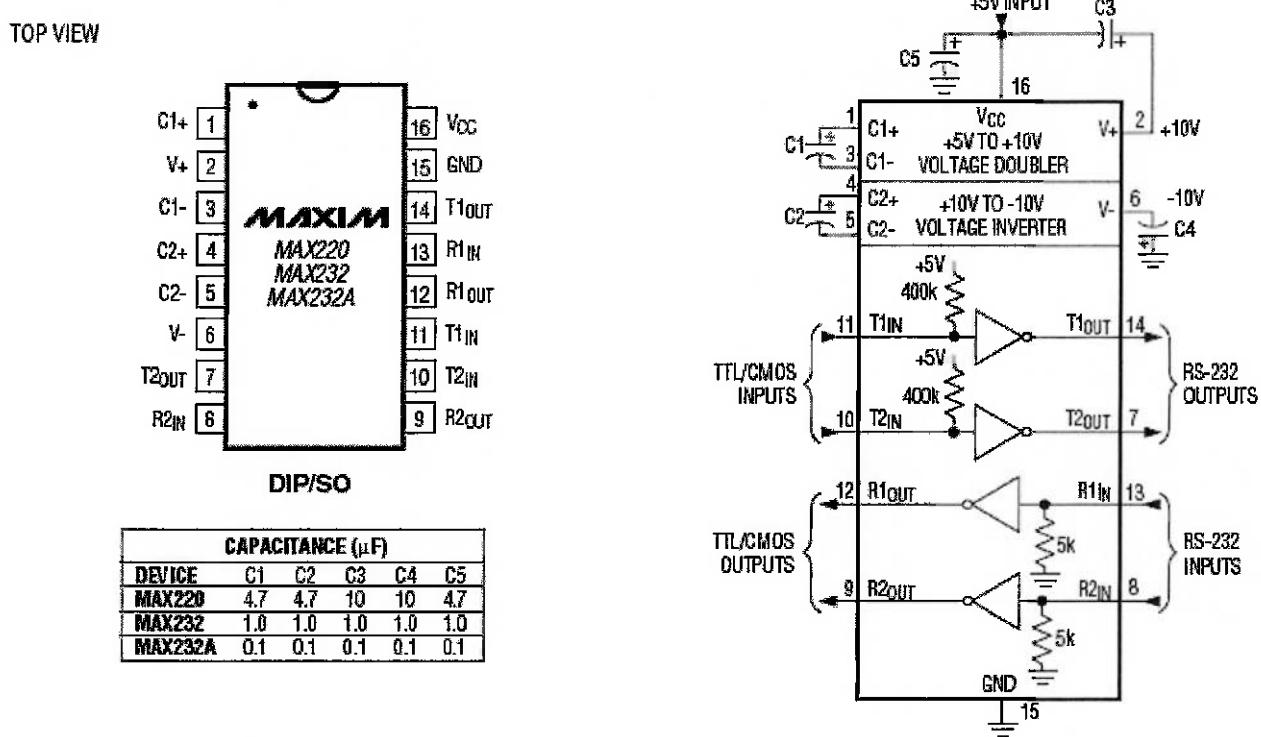
O software que desempenha essa função no projeto é o MySQL. Este é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language* - Linguagem de Consulta Estruturada) como interface.

Este software apresenta algumas vantagens perante a outras ferramentas de base de dados por ele é um software livre que tem uma boa estabilidade e desempenho, visto que os algoritmos que definem as suas operações são otimizados. Além disso, é suportado em uma grande gama de plataformas e é compatível com uma grande variedade de linguagens de programação (C, C++, Java e etc.), nos quais seus os *drivers* podem ser instalados. [Wikipedia]

4.7 Hardware

4.7.1 Max232

Para se modular a comunicação entre o PIC e a porta serial do computador devemos regular as tensões de saída do nosso diapositivo com a tensão de entrada com que a porta serial do microcomputador opera. Para isso utilizamos um conversor de sinais RS232/TTL. Este papel é desempenhado pelo CI MAX232 que possui as seguintes ligações internas.



**Figura 9 – Esquema de funcionamento do MAX232
 [MAX232] Datasheet**

4.7.2 Osciladores

O circuito do oscilador é usado para fornecer um sinal de relógio (*clock*), ao micro controlador. O *clock* é necessário para que o micro controlador possa executar um programa ou as instruções de um programa.

Existem diversos tipos de osciladores. O PIC16F877 pode trabalhar com quatro configurações de oscilador. Uma vez que as configurações com um oscilador de cristal e resistência condensador (RC) são aquelas mais freqüentemente usadas

Quando o oscilador é de cristal, a designação da configuração é de XT, se o oscilador for uma resistência em série com um condensador, tem a designação RC. Isto é importante porque há necessidade de optar entre os diversos tipos de oscilador, quando se escolhe um micro controlador.

No nosso caso foi utilizado um oscilador de cristal. O oscilador de cristal está contido num invólucro de metal com dois pinos onde foi escrita a frequência a que o cristal oscila. Dois condensadores cerâmicos devem ligar

cada um dos pinos do cristal à massa. Existem situações em que cristal e condensadores estão contidos no mesmo "encapsulamento"^a, é o caso do "ressonador" cerâmico. Este elemento tem três pinos com o pino central ligado à massa e os outros dois pinos ligados aos pinos OSC1 e OSC2 do micro controlador

Quando projetamos um dispositivo, a regra é colocar o oscilador tão perto quanto possível do micro controlador, de modo a evitar qualquer interferência nas linhas que ligam o oscilador ao micro controlador.

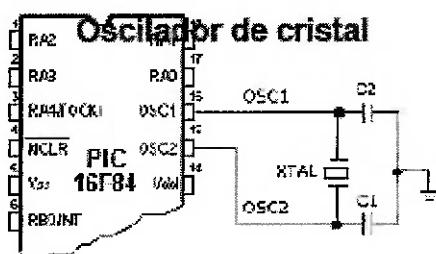


Figura 10 – Oscilador de cristal

Qualquer que seja o oscilador que se está a utilizar, a freqüência de trabalho do micro controlador é a do oscilador dividida por 4. A freqüência de oscilação dividida por 4 também é fornecida no pino OSC2/CLKOUT e pode ser usada para testar ou sincronizar outros circuitos lógicos pertencentes ao sistema

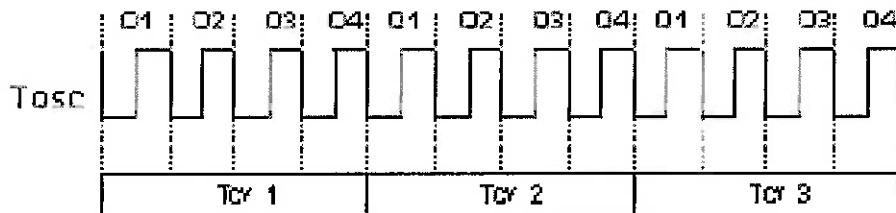


Figura 11 – Forma de onda de oscilador

Ao ligar a alimentação do circuito, o oscilador começa a oscilar. Primeiro com um período de oscilação e uma amplitude instáveis, mas depois de algum tempo tudo estabiliza. Para evitar que esta instabilidade inicial do *clock* afete o funcionamento do micro controlador, necessitamos de manter o micro controlador no estado de *reset* enquanto o *clock* do oscilador não estabiliza

A Figura abaixo mostra uma forma típica do sinal fornecido por um oscilador de cristal de quartzo ao micro controlador quando se liga a alimentação.

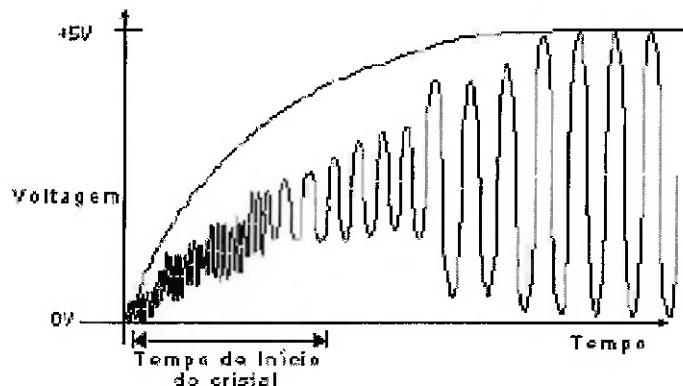


Figura 12 – Interferência de inicio de oscilação [ESTV]

4.7.3 Modulo RF

Os módulos utilizados têm alcance de até 100m sem obstáculos, desde que a antena e a fonte de alimentação do transmissor sigam as recomendações técnicas do fabricante. Esses módulos já saem de fábrica regulados através de tecnologia à Laser, dispensando qualquer tipo de regulagem por parte do desenvolvedor.

Os módulos trabalham nas faixas de freqüências de: 315MHz, 418MHz e 433,92MHz. Os módulos utilizados para criar nosso projeto trabalham na freqüência de 433,92MHz. Pode-se escolher usar qualquer módulo com uma das freqüências citadas acima, mas o par RF deve ter freqüências idênticas para que possa haver uma comunicação entre o transmissor e receptor.

A largura de banda (para a transferência dos dados) do módulo emissor é de 4KHz, já a do módulo receptor é de 2KHz. Portanto, para que o módulo receptor consiga receber os dados corretamente, o transmissor deverá se limitar a transmitir os dados numa taxa inferior ou igual a 2KHz. Veja mais detalhes sobre o modulo RF utilizado no Capítulo 6.6.2.

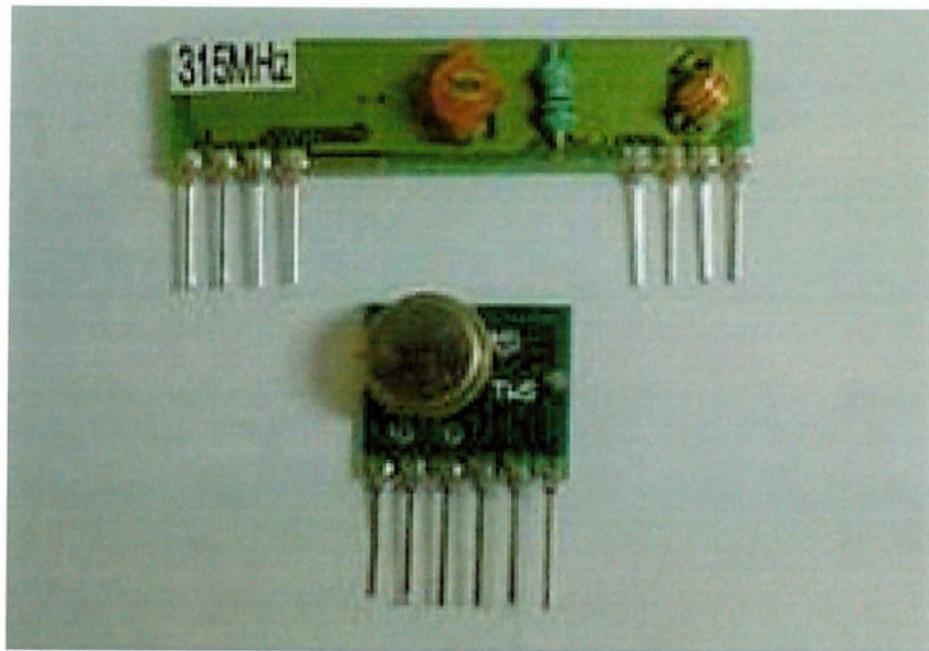


Figura 13 – Foto módulo RF

5 Metodologia

5.1 Análise

Este capítulo tem como objetivo apresentar o estudo de viabilidade feito no inicio do desenvolvimento do projeto, este consistia na verificação da homologação do projeto e da identificação de todos os requisitos necessários para que os objetivos fossem atingidos para que pudéssemos definir qual seria a solução implementada.

5.1.1 Estudo de viabilidade

Inicialmente com um estudo sobre quais são os produtos já utilizados no mercado, características técnicas, preços e disponibilidade analisamos a viabilidade do projeto. Isso impôs a necessidade de se comparar às características destes equipamentos frente a nossa aplicação.

Nesta avaliação, para determinarmos que especificação melhor se adequa ao sistema que adotaríamos, tomamos como base dois modelos: modulo de RF e RFID, tendo em vista que ambos podem tornar essa identificação remota possível. Através da comparação de diversas características, impostos pelas limitações do projeto, observamos aspectos não só econômicos, mas também itens como as possibilidades futuras, tendências e compatibilidades.

Entretanto consideramos também que, tendo em vista o cunho social de nossa aplicação, a simplicidade na implementação e a facilidade para se adotar o mecanismo têm grande importância em nossa estrutura de viabilidade. Isto faz com que os itens custo, a adequação com itens existentes e a padronização para futuros incrementos no sistema montado tenham grande peso para a tomada de uma decisão adequada.

Na Produção do equipamento temos uma estimativa dos custos envolvidos com os diversos itens que o compõe. Em um sistema baseado na tecnologia do RFID temos basicamente as antenas, microprocessadores e

Tags. Enquanto que para o módulo de RF deve-se concentrar em diversos componentes menores: receptor, transmissor, oscilador e outros itens que serão descritos.

Entre os fatores avaliados os mais relevantes são:

- Mais ou menos difundido, a tecnologia que possui mais trabalhos publicados, sendo assim maior a amplitude de pesquisa disponível.
- Facilidade na aplicação, pesquisas realizadas sobre implementações bem sucedidas com a tecnologia em questão.
- Alcance, determinou-se, devido as barreiras naturais (chuvas) e artificiais (carros, proteção metálica), que o alcance deve ser maior que 35m.
- Custos, se os custos de implementação do protótipo, ou mesmo, de um sistema real são viáveis ou não.
- Inclusão tecnológica, uma análise sobre os avanços tecnológicos que serão trazidos com a pesquisa.

O sistema RF já é muito difundido no mercado, é um sistema que já está concretizado com muitas pesquisas sobre e com inúmeras utilizações bem sucedidas de pequenos médios e grande porte.

Uma dificuldade encontrada na utilização de RFID será com certeza o baixo alcance obtido por este (maiores alcances 27m livre, antenas comuns 30cm), isto agravando-se ainda mais pelo fato da necessidade de caixas de proteção para as antenas e a grande quantidade de obstáculos com outros carros, ônibus, chuva, árvores. O RF possui alcance muito maior, como o caso de sistema AM e FM, mas no caso dos controles de alarmes temos um alcance de 40m em média, portanto atendendo as especificações.

A viabilidade do ponto de vista da inclusão tecnológica desse sistema na estrutura de transporte, é também afetada pela atribuição do modelo a ser implantado. A elaboração dos módulos de rádio freqüência em relação à fase de desenvolvimento do RFID, atualmente, é uma estrutura consolidada. Isto pode ser uma vantagem tendo em vista o prazo de duração estimado longo em que o esquema permanecerá montado. Os protocolos de comunicação, por sua vez, podem ser analisados em ambos os sistemas. Todos os aparelhos devem ser homologados e seu protocolo de comunicação deve compatibilizar com os demais terminais com os quais as

informações forem transferidas. Assim esta padronização não deve limitar a projeção do sistema a ser implementado, visto que isto não é um fator limitante na adição de novos equipamentos que possam se integrar à estrutura existente.

5.1.2 Vantagens e Desvantagens

Com base nas considerações mencionadas, determinamos que a viabilidade aponta à utilização dos módulos de radio freqüência para o desenvolvimento de nosso projeto.

Isto foi ponderado de acordo com as informações abaixo na forma de uma tabela com um comparativo sobre RF e RFID nos quesitos relevantes, mostrando as vantagens e desvantagens de cada tecnologia.

Tabela 3 - Vantagens RF X RFID

Vantagens	
RF	RFID
Menor Custo	Protocolos Estabelecidos
Mais Difundido	Tendência
Fácil Aplicação	Inclusão Tecnológica
Grande Alcance	

Tabela 4 - Desvantagens RF X RFID

Desvantagens	
RF	RFID
Instalação	Alto Custo das Antenas
Inclusão Tecnológica	Modelo em Desenvolvimento
Malha Aberta	Baixo Alcance
	Difícil Implementação

5.1.3 Custos Comparativos

Com o ímpeto de reduzir o custo de implementação deste sistema, o que determina a sua viabilidade frente as sua característica como uma ferramenta de auxílio em uma aplicação social, notamos que o volume dos itens a serem incorporados ao sistema tem grande importância para a

validação e aprovação do sistema a ser implementado. Desta forma, devido ao alto custo unitário da antena de RFID de longo alcance, determinamos que este é um componente de alta críticidade na determinação do plano de custos desejado.

A seguir apresentamos a estimativa inicial da aplicação destes dois modelos na cidade de São Paulo e também para a implementação na Cidade Universitária. Devemos frisar que nesta estimativa não são consideradas as plataformas na qual esse sistema será montado. Entretanto, com esse custo é comum a ambos os cenários, e diferença não se altera. A inclusão dessa plataforma será abordada mais a frente neste trabalho, entretanto verificamos que a aplicação desse não é um limitante ao nosso estudo de viabilidade.

Tabela 5 - Estimativa dos Custos de Implementação

Implementação de RFID no Sistema em uma linha padrão em SP						
Número de pontos por linha	Custo da antena	Custo do "tag"	Número de ônibus por linha	Total (R\$) por linha	Total (R\$) na cidade de SP	Total por linha considerando periféricos
35	R\$ 2.000,00	R\$ 15,00	14	R\$ 70.210,00	R\$42.126.000,00	R\$ 71.960,00
Implementação do RFID em uma linha do circular da USP						
Número de pontos por linha	Custo da antena	Custo do "tag"	Número de ônibus por linha	Total (R\$) por linha	Total (R\$) na Cidade Universitária	Total por linha considerando periféricos
25	R\$ 2.000,00	R\$ 15,00	4	R\$ 50.060,00	R\$ 100.120,00	R\$ 51.310,00
Implementação de RF no Sistema em uma linha padrão em SP						
Número de pontos por linha	Custo do módulo no ponto	Custo do módulo no ônibus	Número de ônibus por linha	Total (R\$) por linha	Total (R\$) na cidade de SP	Total por linha considerando periféricos
35	R\$ 150,00	R\$ 50,00	14	R\$ 5.950,00	R\$ 3.570.000,00	R\$ 7.700,00
Implementação do RF em uma linha do circular da USP						
Número de pontos por linha	Custo da antena de RF	Custo do módulo no ônibus	Número de ônibus por linha	Total (R\$) por linha	Total (R\$) na Cidade Universitária	Total por linha considerando periféricos
25	R\$ 150,00	R\$ 50,00	4	R\$ 3.950,00	R\$ 7.900,00	R\$ 5.200,00

5.1.4 Legislação e Homologação

Tendo em vista que nosso equipamento lida com ondas eletromagnéticas, devemos atender as normas da Anatel (Agencia Nacional de Telecomunicações) para a utilização da freqüência em que o equipamento de nosso protótipo opera. Para que possamos também oficializar o uso deste produto devemos atender as normas regidas para a homologação junto a Anatel. Podemos verificar tal regulamentação no anexo deste trabalho. As tabelas abaixo mostram as faixas de freqüência e potencia restritas as quais estamos submetidos na formulação de nosso sistema.

Tabela 6 - Tabela de faixas de freqüências com restrição de uso

Tabela I
Faixas com restrições de uso

MHz	MHz	MHz	GHz
0,090-0,110	13,36-13,41	399,9-410	5,35-5,46
0,495-0,505	16,42-16,423	608-614	6,65-6,6752
2,1735-2,1905	16,69475-16,69525	952-1215	8,025-8,5
4,125-4,128	16,80425-16,80475	1300-1427	9,0-9,2
4,17725-4,17775	21,87-21,924	1435-1646,5	9,3-9,5
4,20725-4,20775	23,2-23,35	1660-1710	10,6-11,7
6,215-6,218	25,5-25,67	1718,8-1722,2	12,2-12,7
6,26775-6,26825	37,5-38,25	2200-2300	13,25-13,4
6,31175-6,31225	73-74,6	2483,5-2500	14,47-14,5
8,291-8,294	74,8-75,2	2655-2900	15,35-16,2
8,362-8,366	108-138	3260-3267	20,2-21,26
8,37625-8,38675	149,9-150,05	3332-3339	22,01-23,12
8,41425-8,41475	156,52475-156,52525	3345,8-3352,5	23,6-24,0
12,29-12,293	156,7-156,9	4200-4400	31,2-31,8
12,51975-12,52025	242,95-243	4800-5150	36,43-36,5
12,57675-12,57725	322-335,4		Acima de 38,6

Tabela 7 – Tabela de limites gerais de emissão.

Faixa de freqüências (MHz, onde não especificado)	Intensidade de campo (microvolt por metro)	Distância da Medida (metro)
9-490 kHz	2400/F(kHz)	300
490-1705 kHz	24000/F(kHz)	30
1,705-30	30	30
30-88	100	3
88-216	150	3
216-960	200	3
Acima de 960	500	3

5.2 Projeção

Com os requisitos devidamente traçados podemos passar para a fase de desenvolvimento. Nesta etapa geramos algumas opções que atendem as necessidades impostas pelo sistema de identificação de linhas de ônibus. Analisamos o uso de tecnologias em radio freqüência como RFID (*Radio Frequency Identification*) e módulos de RF com os quais poderíamos nos basear. A partir disso, definimos a tecnologia de módulos de radio freqüência que melhor se encaixava a nossa aplicação e que seria essa a nossa ferramenta de desenvolvimento. Deste poderíamos nos dedicar ao desenvolvimento, que será abordado mais detalhadamente nos capítulos seguintes, a fim de detalhar o projeto. Este relatório descreve a abordagem completa dessa etapa.

5.3 Construção

Como resultado, após atingir o detalhamento necessário, construímos um protótipo que deve ser capaz de materializar o modelo desenvolvido. Esse, leva para campo o modelo proposto que serve como ferramenta de auxílio aos deficientes visuais, através da identificação das linhas de ônibus por aviso sonoro. Ao longo deste relatório as atividades desenvolvidas na constituição do protótipo também são descritas, tendo em vista o cumprimento da meta estabelecida por este projeto.

6 Projeto

Este capítulo tem como objetivo apresentar todas as informações relevantes ao projeto, como os esquemas elétricos, componentes utilizados, custos e códigos de programação, visando assim a completa descrição do projeto e através dos modelos adotados.

6.1 Introdução

Inicialmente apresentaremos o projeto de uma forma mais generalista. O sistema tem como componente estrutural básico um par de moduladores RF, um transmissor e um receptor. A fim de melhorar o entendimento do sistema, este será dividido em três partes.

Emissor: Este sistema é responsável por emitir sinais de rádio modulando um número ID que identifica a linha do ônibus. O emissor consiste basicamente de um emissor RF que recebe um sinal serial (Código ID) enviado um processador PIC via transmissão serial. Para minimizar os erros de sincronia de recepção e emissão foi necessário colocar um oscilador o qual são utilizados os sinais de *clock* a fim de sincronizar o emissor e o receptor. Como o emissor é fixado dentro de um ônibus utiliza-se a própria bateria do ônibus colocando em série com um regulador de tensão de 12V → 5V. Para este software foi utilizado programação em C e as bibliotecas do compilador MikroC.

Receptor: Este sistema possui um receptor RF integrado com um micro computador. Este transmite o sinal para o micro computador via serial. Através de análise de interrupção, os sinais vindos do RF são interpretados, filtrados (eliminação de ruídos) e finalmente gravados em uma memória. Sendo assim enviado para a terceira parte do sistema um banco de dados. Para este receptor também é utilizado um regulador e como é fixado em um ponto de ônibus (120AC) foi necessário um transformador e um conversor AC-DC. Para este software foi utilizado programação em C++.

Banco de Dados: O Programa recebe os valores (ID) e faz um SELECT em um banco de dados. Assim faz todas as análises logísticas

necessárias e obtém o endereço para tocar o som referente à linha de ônibus identificada. Para este software foi utilizada a linguagem de programação C++. Com o endereço do arquivo *.wav definido o programa executará o som em caixas amplificadoras.

Veja abaixo um esquema geral do sistema.

Esquema Geral

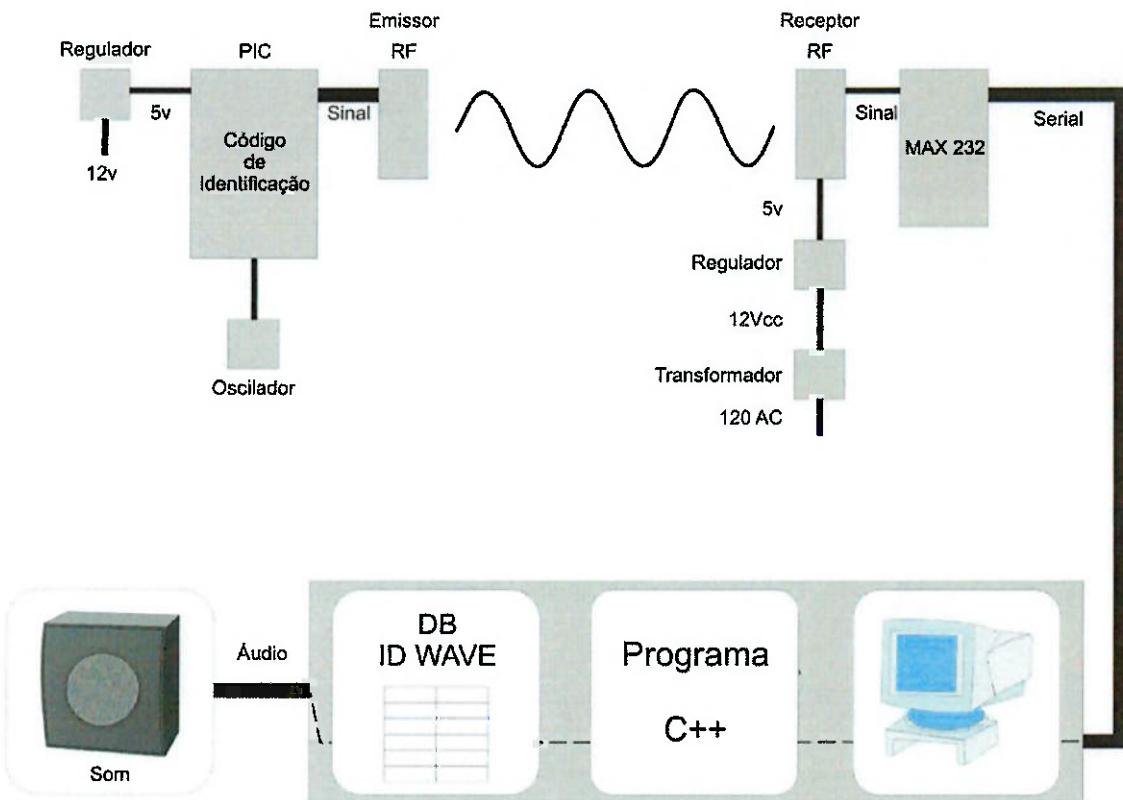


Figura 14 – Esquema geral de funcionamento

6.2 Alterações no Projeto

Ao decorrer do projeto algumas alterações devido a necessidades construtivas e de redução de custos, foram necessárias.

As alterações menos relevantes do projeto, como modificação de capacitores, resistores ou outro componente de menor importância não serão discutidos neste capítulo, pois neste documento encontram-se os resultados finais do projeto e não julgamos interessante para o leitor saber destas alterações do projeto.

A primeira alteração relevante do projeto foi, devido aos resultados obtidos com o estudo de viabilidade, utilizar um módulo RF ao invés de um par RFID. Os motivos pelos qual esta decisão foi tomada já foram discutidos com mais detalhes no capítulo 5.1. Esta alteração teve como objetivo principal reduzir drasticamente os custos do projeto, tornando-o viável para uma possível implementação.

A segunda alteração estrutural do projeto foi a remoção do segundo PIC, proposto no projeto inicial (Ver Relatório Parcial). Esta alteração foi realizada, pois identificamos que o micro poderia se comunicar via radiofrequência diretamente com o PIC. Sendo assim, está foi obviamente uma redução de custos e paralelamente reduziu as chances de ocorrer erros no sistema, tornando-o mais robusto. Com esta alteração as interpretações de protocolos e sincronia da comunicação por PIC foram feitas diretamente pelo micro computador. Este por sua vez, também faz a conexão com o banco de dados e toca o som com as informações necessárias.

Na parte de software, tiveram duas alterações. Primeiramente foi alterada a linguagem de programação base do sistema de Java para C++. O motivo para esta alteração foi a maior afinidade com a segunda linguagem, pode-se levar em consideração também que C++ é uma linguagem mais robusta. Outra alteração muito importante ocorrida foi o compilador de linguagem Assemby. Inicialmente estávamos utilizando o PIC-C, com este não conseguíamos obter os resultados certos e até pensamos em alterar o gravador. Após inúmeros testes com outros compiladores encontramos um que fosse compatível com o gravador de PIC, MICKO-C. Com este compilador os resultados se concretizaram gerando um grande desenvolvimento para o projeto.

6.3 Características Técnicas

Para melhor entendimento e organização este capítulo será dividido em dois subitens, emissor e receptor.

6.3.1 Emissor

O emissor é controlado por um PIC 16F877 (Ver detalhes no capítulo 6.61). Este é alimentado em VCC nos pinos 11 e 32, ligado ao terra nos pinos 12 e 31, utiliza-se o pino 1 como um botão de *reset*, para isso ligamos a VCC em serie com um resistor de 1KΩ.

O PIC recebe sinais de *clock* através dos pinos 13 e 14 de um oscilador de cristal de quartzo de 10MHz, este é ligado a dois capacitores e ao terra (Ver detalhes no capítulo 6.6.3).

Com a freqüência de oscilação regulada, o PIC envia os dados através da porta serial, pino 25, com um taxa de transmissão de 1200bps, sem bit de paridade.

Estes dados vão para um emissor de radiofreqüência que trabalha com freqüência de 433,92MHZ. (Ver detalhes no capítulo 6.6.2)

6.3.2 Receptor

O receptor é controlado por um micro computador ou um palmtop com Windows instalado. A comunicação é feita por uma porta serial COM2. Esta está ligada a um conversor RS232 / TTL no pino 7 do componente MAX232 (Ver detalhes no capítulo 4.6.5).

Para receber os sinais o MAX232 está ligado através do pino 8 ao receptor de radiofreqüência que trabalha com a freqüência de 433,92MHZ. (Ver detalhes no capítulo 6.6.2)

Os sinais recebidos pelo receptor de radiofreqüência são então enviados para o MAX232 que é ampliado de 5V para 12V. Este por sua vez, se conecta a porta serial do micro computador e é lido por um programa leitor de serial (Ver detalhes no capítulo 6.11.2). Este verifica se os dados recebidos obedecem aos protocolos estabelecidos. No caso afirmativo envia os dados para um gerenciador de banco de dados (Ver detalhes no capítulo 6.11.3).

O gerenciador de banco de dados recebe a string com valores a serem manipulados, verifica se estes existem no banco, no caso afirmativo realiza operações de logística para verificar se o dado não é repetido e finalmente

seleciona no banco de dados o endereço que está armazenado o som para aquela linha de ônibus. Finalizando do o processo o programa executa o som. (Ver detalhes no capítulo 6.11.4)

Veja todas as ligações no capítulo abaixo.

6.4 Esquema Completo de Funcionamento

Veja abaixo o circuito do emissor

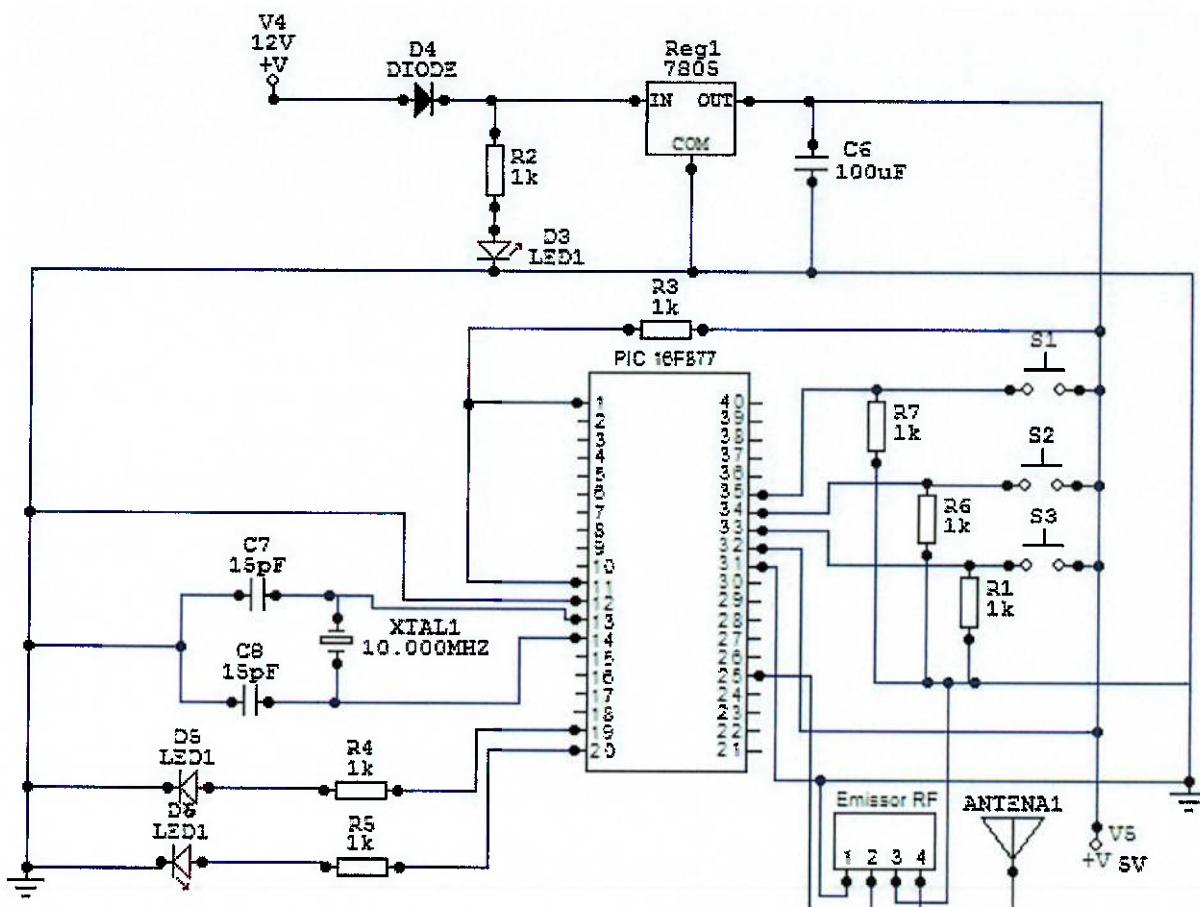


Figura 15 -- Esquema de ligação do emissor

Veja abaixo o circuito completo do receptor

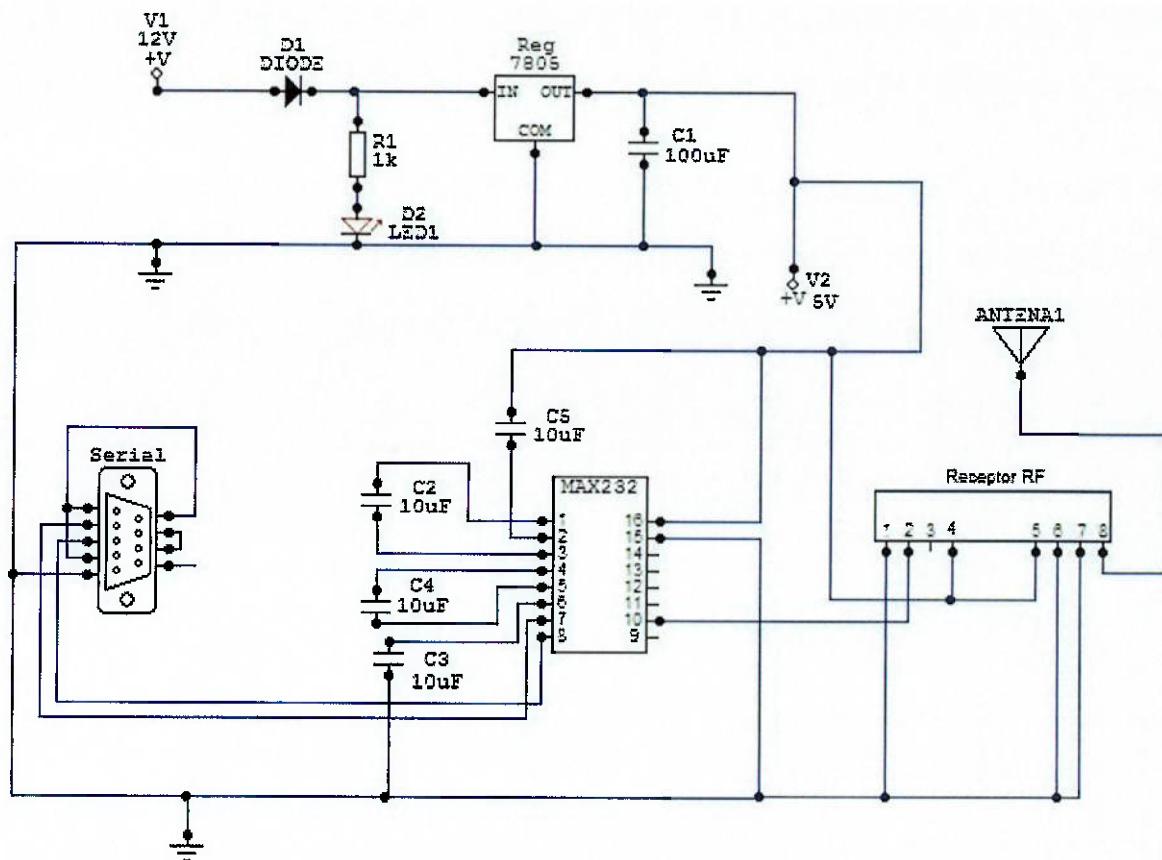


Figura 16 – Esquema de ligação do receptor

6.5 Custo do Protótipo

6.5.1 Dispositivo móvel (emissor)

O preço total de cada dispositivo é:

- Micro controlador PIC – R\$27,10
- Emissor de radiofreqüência – R\$15,00
- Regulador de Tensão – R\$5,30
- Osciladores – R\$1,70
- Placa – R\$2,50
- **EMISSOR – R\$51,60**

6.5.2 Receptor

Para o receptor não levamos em consideração aqui o custo de um computador (plataforma), visto que esses não foram objetos de estudo de nosso projeto. Contudo está aplicação será discutida nos resultados desse trabalho.

O preço total de cada dispositivo é:

- Receptor de radiofreqüência – R\$15,00
- Regulador de Tensão – R\$5,30
- MAX 232 – R\$4,00
- Osciladores – R\$1,70
- Placa – R\$2,50
- **RECEPTOR – R\$28,50**

6.6 Componentes Utilizados

Este capítulo tem como objetivo descrever detalhadamente cada dispositivo, descrevendo os motivos da escolha, preços, fotos, e circuito necessários.

6.6.1 Micro controlador PIC

O micro controlador PIC utilizado foi o PIC16F877. Este é um processador para protótipo, pois ele tem uma memória flash que pode ser re-grada quantas vezes necessárias, o custo médio deste dispositivo é de vinte e sete reais (R\$27,00).

A função do dispositivo é controlar todo o envio de dados via porta serial e radiofreqüência dos dados (ID) de identificação do veículo. Este dispositivo é de fundamental importância para o funcionamento do sistema, nele é gravado o número de identificação e ele também controla todos os protocolos e freqüências de comunicação. Para este dispositivo utiliza-se um resistor em serie de $1K\Omega$ na alimentação VCC.

O principal motivo de escolha deste micro controlador é fato dele possuir uma interface serial pré-programada, ter baixo consumo de energia elétrica ter freqüência de oscilação máxima superior a 10MHz, possuir *pull-up* na porta B e gera interrupções automáticas quando há algum dado em espera na porta serial. Veja abaixo a descrição completa de todas as características deste micro controlador.

- RISC CPU de alta performance.
- Velocidade de operação de até: DC - 20 MHz *clock* input
- Até 8K x 14 palavra na memória flash
- Até 368 x 8 bytes de memória de dados (RAM)
- Até 256 x 8 bytes de memória EEPROM
- *Pinout* compatível com PIC16C73B/74B/76/77
- Possibilidade de interrupções (até 14 fontes)
- Oito níveis hardware *stack*
- Modo de endereçamento direto, indireto e relativo
- *Power-on Reset* (POR)
- *Power-up Timer* (PWRT) e *Oscillator Start-up Timer* (OST)
- *Watchdog Timer* (WDT)
- Código de proteção de programa
- Economia de energia, modo SLEEP
- Osciladores selecionáveis
- Tecnologia de baixo consumo e alta velocidade CMOS

FLASH/EEPROM

- Programação serial (ICSP) através de dois pinos
- Programação serial de 5V
- *Debugging* apenas por dois pinos
- Processador lê/escreve acessos para memória de programa
- Corrente disponível: 25 mA
- Comercial, Industrial faixas de temperatura
- Baixo consumo de potência:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 μ A typical @ 3V, 32 kHz

- $< 1 \mu\text{A}$ *typical standby current*
- Timer0: 8-bit timer/counter com 8-bit prescaler
- Timer1: 16-bit timer/counter com prescaler, pode ser incrementado durante o SLEEP mode via clock externo.
- Timer2: 8-bit timer/counter com 8-bit register, prescaler and postscaler
 - Duas Capturas, Comparações, PWM módulos
 - Captura é 16-bit, max. resolução é 12.5 ns
 - Comparaçao é 16-bit, max. resolução é 200 ns
 - PWM max. resolução é 10-bit
- 10-bit multi-canal Analogico-para-Digital conversor
- Porta serial sincrona (SSP) com SPI (Master mode) e I2C (Master/Slave)
 - Universal Síncrono e Assíncrono receptor Transmissor (USART/SCI) com 9-bit.
 - Porta paralela (PSP) 8-bits, com externo RD, WR and CS controle (40/44-pin only)

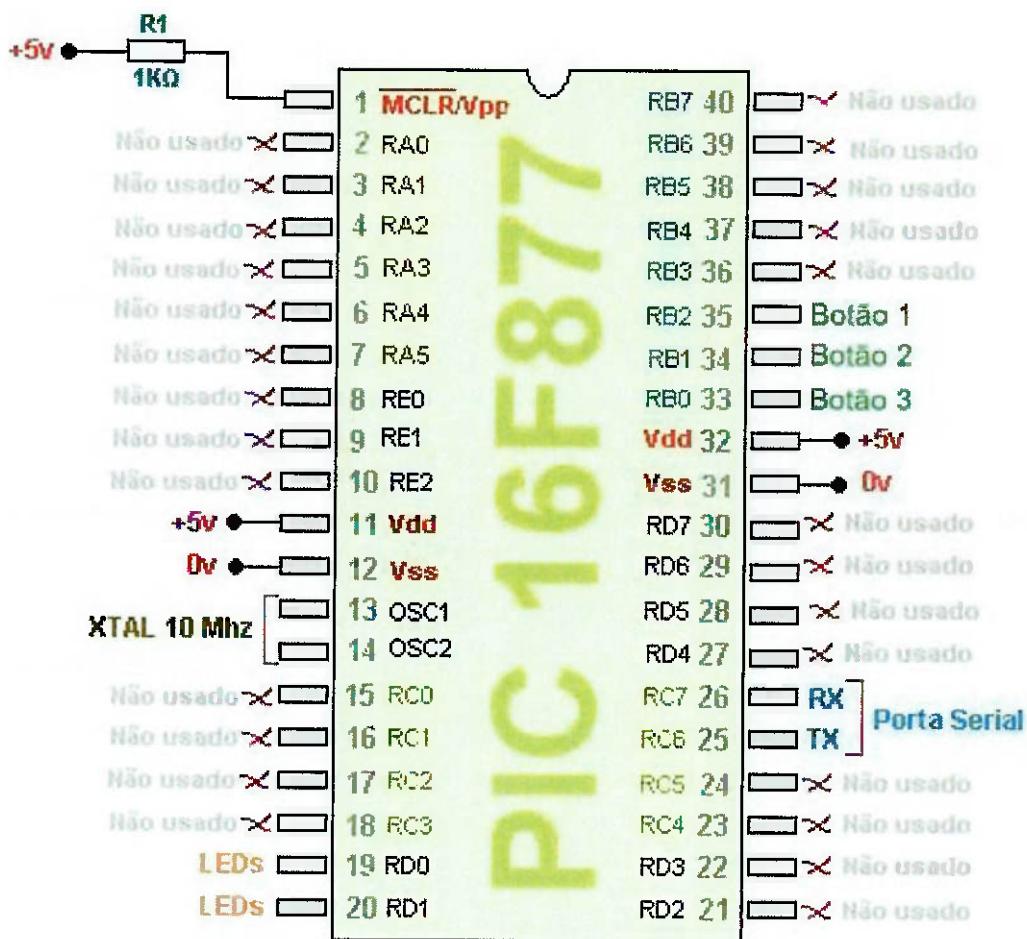


Figura 17 – Ligações do PIC

6.6.2 Modulo de Radiofreqüência

O módulo de rádio freqüência tem a função de fazer a ligação entre o emissor e o receptor. Os sinais são recebidos pelo emissor via o pino 2, envia os dados por radiofreqüência e estes são recebidos pelo receptor RF que após passar por filtros e amplificadores são colocados no pino 2. Veja maiores detalhes sobre o funcionamento do módulo RF e suas respectivas ligações.

6.6.2.1 Freqüência Utilizada

Para determinar a freqüência a ser utilizada primeiramente foi feito um estudo sobre as freqüências disponíveis na ANATEL. Ver detalhes no capítulo 5.1.4.

Com base nestes estudos foi escolhida a freqüência $f = 433,92\text{MHz}$.

Para esta freqüência foi feito alguns estudos complementares, assim verificando a viabilidade técnica desta.

6.6.2.1.1 Efeito Doppler

O sistema RF estudado será submetido a movimentos relativos entre o transmissor e o receptor, sendo assim foi necessário verificar se o efeito Doppler iria causar alguma variação relevante na freqüência do sinal RF.

O efeito Doppler consiste em “para um corpo luminoso se aproximando (ou se afastando) do observador o comprimento de onda da luz diminui (aumenta), em relação àquele observado em laboratório”. Isto é, para um corpo em movimento a frequencia emitida por um corpo fixo é diferente da frequencia recebida pelo corpo em movimento e vice e versa.

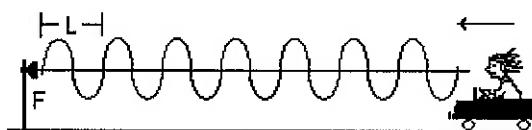


Figura 18 - Efeito Doppler

A frequencia recebia é calculada por: [Amperes]

$$f' = f + v_o/L$$

Sendo L o comprimento de onda. Assim:

$$L = c/f_0$$

Para a freqüência em estudo $433,92\text{MHz}$ temos que $L = 0,69137\text{ m}$

Para uma velocidade máxima de $19,5\text{m/s}$

$$f' = 433.920.000 + 28,1245\text{ Hz} = 433.920.28,1245\text{Hz}$$

Assim a variação máxima dá freqüência devido ao efeito Doppler é $6,48 \times 10^{-6}\%$ e, portanto não vai prejudicar a transmissão.

6.6.2.2 Características do Módulo RF

Para melhor entendimento das características do módulo, este foi dividido em duas partes, receptor e emissor.

6.6.2.2.1 Receptor RF

O receptor é responsável por receber os sinais via radiofreqüência, ele trabalha com a freqüência de 433,92Hz.

O receptor funciona basicamente da seguinte forma. Os dados são captados pela antena, em seguida passa por um oscilador onde é regulada a freqüência de transmissão, logo após estes sinais passam por um filtro onde são eliminado os ruídos. Com os ruídos eliminados o sinal passa por um amplificador atingindo a tensão de 5V, e finalmente passa por um comparador. Veja abaixo um esquema dos processos pelos quais os sinais captados passam. [RF Datasheet]

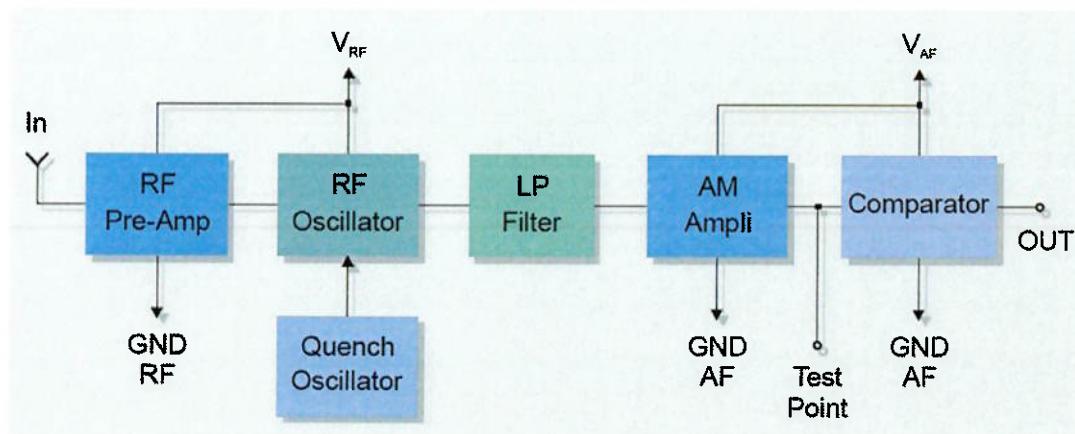


Figura 19 – Diagrama de blocos do receptor RF

Este receptor se liga com o resto do circuito da seguinte forma. Os pinos 1, 6 e 7 são ligados ao terra, os pinos 4 e 5 são ligados à VCC, o pino

8 é ligado a antena e o pino 2 é ligado a entrada de sinais RX do MAX232
Pino 8 (Ver detalhes capítulo 6.6.5) e assim os dados são transmitidos ao do
micro computador via comunicação serial.

Cada emissor custa R\$15,00.

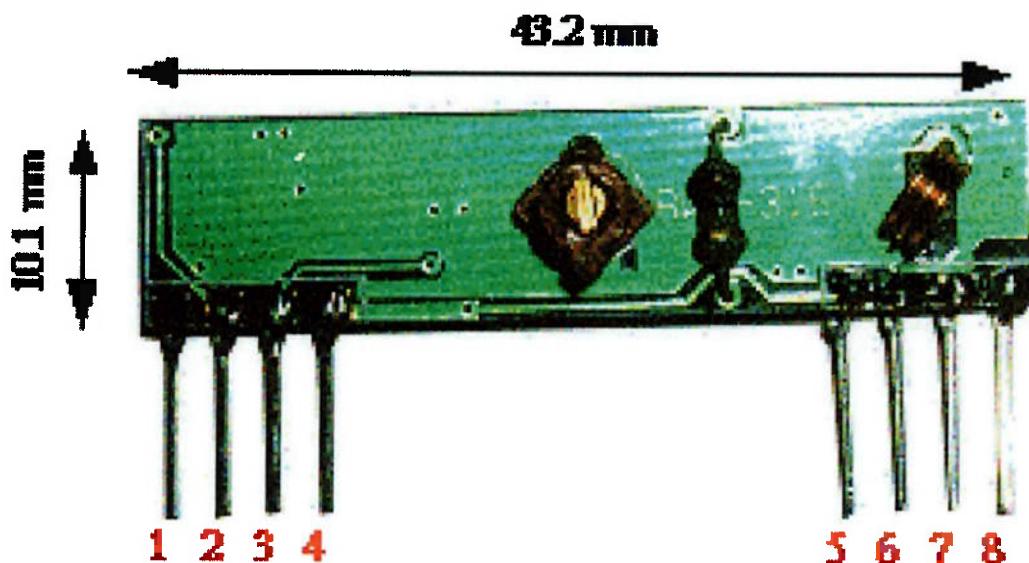


Figura 20 – Ligação do receptor RF

6.6.2.3 Emissor

O emissor é responsável por enviar os dados. Veja abaixo o esquema elétrico do emissor. [RF Datasheet]

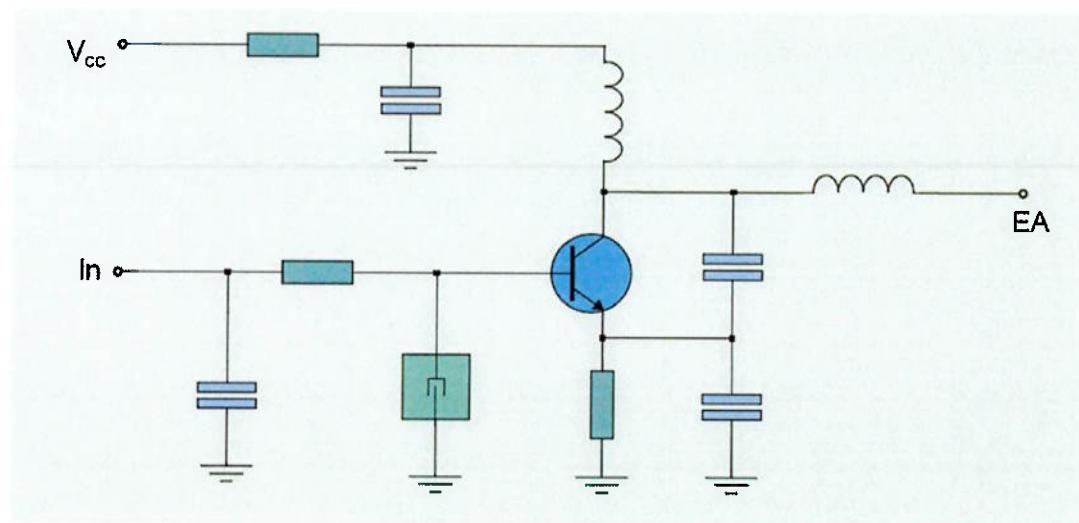


Figura 21 – Esquema elétrico do emissor

Este emissor se liga com o resto do circuito da seguinte forma. O pino 1 é ligado ao terra, o pino 3 é ligados à VCC, o pino 4 é ligado a antena e o pino 2 é ligado a saída de sinais TX do PIC Pino 25, (Ver detalhes capítulo 6.6.1) e assim os dados são transmitidos ao do micro computador via comunicação serial.

Cada emissor custa R\$15,00.



Figura 22 – Ligações do emissor RF

6.6.3 Osciladores

O oscilador é parte fundamental do sistema, ele é responsável pela sincronia da transmissão de dados (Ver detalhes no capítulo 4.6.2). Basicamente, o oscilador envia sinais de clock para o PIC o qual regula todos os seus processos com estas oscilações.

O PIC utilizado aceita velocidade de até 20MHz, para este sistema utilizou-se a freqüência de oscilação de 10MHz isso porque a transmissão de dados via radiofreqüência é 1200bps e assim o PIC não consegue, dado uma freqüência de 20MHz, reduzir para a oscilação necessária para o envio de dados.

O dispositivo constitui-se basicamente de 2 capacitores de 15pF e um **XTAL**, que nada mais é do que um cristal de quartzo oscilando a 10.000.000 (dez milhões de vezes por segundo) ou simplesmente **10Mhz**. É esse o cristal que gera os pulsos precisos para o bom funcionamento do micro controlador.

O Custo total do dispositivo é R\$1,70. Sendo R\$0,20 os capacitores e R\$1, 50 o cristal de quartzo.

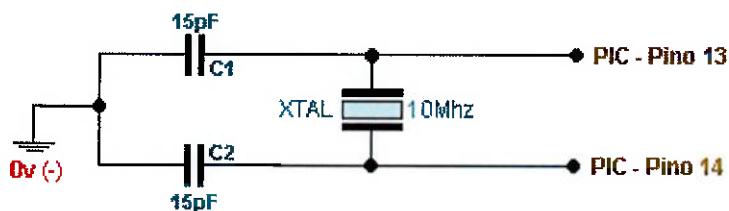


Figura 23 – Ligações do oscilador

6.6.4 Reguladores de Tensão

As fontes convencionais transformam a tensão alternada de 110V ou 220V AC para 12V, mas devido a oscilações na rede elétrica e mesmo a falta de precisão desta transformação da fonte este valor oscila. Esta oscilação prejudica muito o funcionamento do PIC e outros CI, podendo até queimá-los em pico de tensão. Sendo assim, é necessária a utilização de um regulador de voltagem em 5V.

Este dispositivo utiliza um regulador de tensão de 12V para 5V o componente 7805, um capacitor eletrolítico de 100 μ F / 25V um resistor, um diodo 1N4007. Basicamente o sistema funciona da seguinte maneira, aplicamos uma tensão de 12V no pino 1 do regulador, no pino 2 (centro) temos o terra e no pino 3 conseguimos os 5V. O custo total do dispositivo foi R\$5,30, sendo o regulador R\$5,00, o capacitor R\$0,10 o LED R\$0,10 e o diodo R\$0,10.

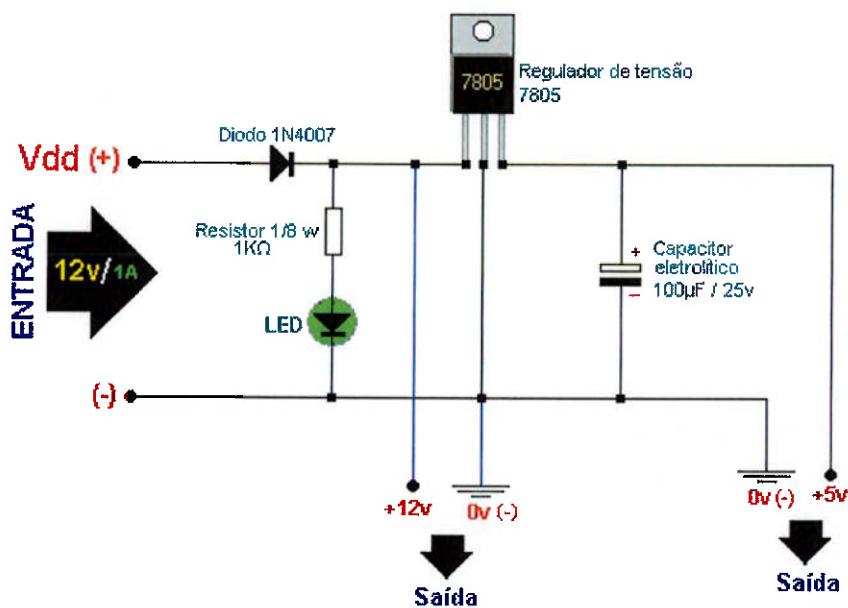


Figura 24 – Ligações do regulador de tensão

6.6.5 Ligação Serial

Como já descrito no capítulo 4.3 a porta serial de micro computador trabalha com 12V então é necessário utilizar um conversor de sinais RS232/TTL, para isso utilizamos o CI MAX232. Como pode ser observado na figura abaixo o pino 7 e o pino 8 são ligados ao RX e TX do PC respectivamente. O pino 9 e o pino 10 são ligados às portas do RX e TX do PIC respectivamente.

Este CI recebe em tensão de VCC igual a 5V e possui quatro capacitores eletrolítico de $12\mu\text{F}$ cada. Este capacitores como objetivo filtrarem a tensão de ruídos a fim de manter os 5V ou 12V, o primeiro para o PIC e o segundo para o PC.

Todo o sistema custou R\$4,00 reais sendo R\$0,50 os capacitores e R\$3,50 o CI MAX232.

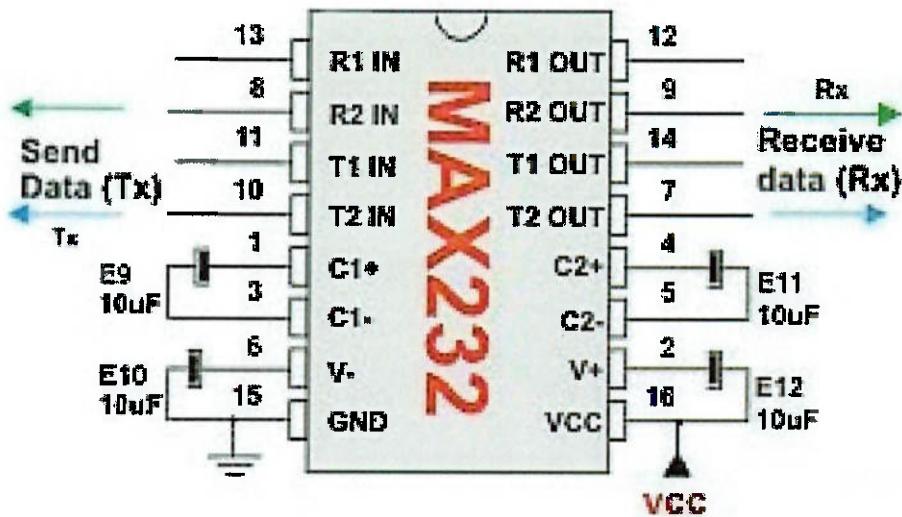


Figura 25 – Esquema de ligação do RS232.

6.7 Protocolo de Transmissão

Uma comunicação via radiofreqüência está sujeita aos ruídos e interferências, isto de fato, pois a faixa de freqüência de trabalho pode ser utilizada por diversos dispositivos de pequeno porte. Sendo assim foi necessário utilizar um protocolo de comunicação.

Analizando os dispositivos disponíveis no mercado, observou-se que muitos os módulos RF trabalham e velocidade de até 2Kbits/s. Esta velocidade é facilmente suportável por um micro controlador que trabalha em MHz. Com base nestas informações decidiu-se em utilizar uma taxa de transferência de 1200 bits/s.

Para uma transmissão eficiente com base em coeficientes de segurança e taxas de erros foi decidido que o tamanho máximo dos dados enviados será de 64bits. Sendo assim os dados de identificação do ônibus mais o protocolo de informação não poderia exceder 64bits.

Sendo assim foi decidido utilizar três chaves de 8bits (1byte) para identificar se o sinal foi transmitido corretamente ou não, sobrando então 5bytes para dados. Em um modelo piloto, que é o protótipo que construído

foi utilizado apenas 4bytes para identificação restando assim 1byte para futuras implementações.

As três chaves utilizadas ficaram dispostas da seguinte forma:

- Chave 1: STX → Chave START, este valor é enviado sempre no começo do trem de dados para indicar que está começando o envio de uma informação. Com base em informações de terceiros o número 0x02 é comumente utilizado e é mais inerte a ruídos.
- Dado 1: IDA → Conjunto A de dados, este valor conterá os bits mais significativos do número de identificação.
- Dado 2: IDB → Conjunto B de dados, este valor conterá os bits menos significativos do número de identificação.
- Chave 2: ETX → Chave END, este valor será enviado sempre logo após as dados, avisando assim o processador que os dados chegaram ao fim. Com base em informações de terceiros o número 0x01 é comumente utilizado.
- Chave 3: SUM → Chave SOMA, este valor contém um valor para verificar se os dados transmitidos foram certos e enviados para a antena certa. Este código soma contém o valor da soma de todos os dados.

Caso uma das três chaves não seja detectada o programa descarta os dados recebidos.

Veja abaixo um esquema ilustrativo do protocolo que será utilizado.

Chave1	Dados1	Dados2	Chave2	Chave3
STX	IDA	IDB	ETX	SUM

Figura 26 – Protocolo de Transmissão

Veja abaixo os principais pontos do código implementado para gerar e checar o protocolo.

A função abaixo define os protocolos de comunicação, para isto zera todos os *flags* de detecção de protocolo e grava em duas Strings o START byte e o END byte.

```
//-----
void DefineProtocolo(void) {
    ProtI = false;                                //Zera os Booleanos do protocolo
    ProtF = false;                                //Zera os Booleanos do protocolo
    ProtSUM = false;                               //Zera os Booleanos do Protocolo
    memset(StringP1, 0, sizeof(StringP1)); //zera string
    memset(StringP2, 0, sizeof(StringP2)); //zera string
    StringP1[0] = 0x02;                           //Escreve protocolo inicial
    StringP1[1] = '\0';
    StringP2[0] = 0x01;                           //Escreve protocolo final
    StringP2[1] = '\0';
}
//-----
```

Código 1 – Código fonte que define o protocolo de comunicação

A função abaixo faz parte do programa de leitura da porta serial, este tem como objetivo identificar se os dados recebidos pela porta serial estão seguindo os protocolos de comunicação estabelecidos. Para isto ela verifica etapa por etapa cada fase da identificação e nos casos positivos, ativa flags que gravam as etapas já identificadas.

```
//-----
void VerificaString(void) {
    char SUM [16];
    int i = 0;
    int cont = 0;
    if(String2 [0] == StringP1[0]) { ProtI = true; printf("\nProt I"); } //Verifica protocolo inicial
    else { ProtI = false; memset(String2, 0, sizeof(String2)); } //caso negativo deleta string
    if(ProtI == true && String2[N_dados + 1] == StringP2[0]) {
        ProtF = true; printf("\nProt F"); //verifica protocolo final
    }
    else {
        ProtI = false;
        ProtF = false;
    }
    if(ProtI == true && ProtF == true && strlen(String2) == N_dados + 3) { //Verifica SUM
        printf("\nEntra Prot SUM");
        for(i = 1; i < strlen(String2)-1; i++) { //calcula SUM
            cont = cont + (int)(String2[i]);
        }
        SUM[0] = (char)(cont);
        SUM[1] = '\0';
        if(String2 [N_dados + 2] == SUM[0]) ProtSUM = true;
        else {
            ProtI = false;
            ProtF = false;
            ProtSUM = false;
        }
    }
}
//-----
```

Código 2 – Código fonte que verifica o protocolo de comunicação

6.8 Sistema de Logística

Como não podemos controlar o número de vezes que o sinal é recebido pelo receptor de um mesmo veículo foi necessário solucionar este erro, para isso foi implementado alguns algoritmos de logística.

6.8.1 Detecção de Sinais repetidos

Como mencionado anteriormente os transmissores RF ficam enviando sinais intermitentemente, sendo assim o receptor RF vai receber o ID do ônibus mais de uma vez. O problema então se dá no fato de filtrar os sinais quando já tiverem sido anunciados.

Como se sabe que o ID será único e exclusivo para todos os ônibus, portanto, o sistema rejeitará se o mesmo ID for recebido dentro de um intervalo de tempo muito pequeno, sabendo que o ônibus para ser detectado novamente precisa completar todo o percurso da linha.

Para isso o sistema irá gravar a última vez que anunciou aquele mesmo veículo e comparar com a nova entrada de dados. Uma outra dificuldade foi encontrada então, o caso de virar o dia. Como estamos comparando horas existe a possibilidade de um ônibus estar exatamente próximo às 24h e logo depois passar às 0h. Isso ocorreria um erro, pois a diferença entre os tempos seria de aproximadamente vinte quatro horas e, portanto o sistema identificaria como um novo ônibus e anunciaria novamente.

Para corrigir este problema foi inserida uma nova cláusula no programa, se a diferença entre os tempos for negativa o programa soma vinte e quatro horas na nova contagem, sendo assim eliminamos o problema. (Ver mais detalhes no capítulo 6.8.1).

6.8.2 Vias de duas mãos

Uma das dificuldades encontradas foi nas vias públicas que existe o transito de veículos nas duas direções. Neste caso o sistema teria que identificar se o ônibus está partindo ou chegando ao ponto, para não seja identificado duas vezes.

Como para todos os pontos será instalado uma antena antes do mesmo, para este caso em particular também será instalado apenas uma antena antes, porém nas duas direções, ficando assim com uma antena antes e outra depois do ponto (uma pra cada direção) ambas controladas pelo mesmo micro computador. O resultado disso é a possibilidade de integração entre estas duas antenas e, portanto, uma análise de logística da situação.

As antenas ficarão dispostas de forma que o ônibus não seja detectado pelas duas antenas ao mesmo tempo, assim o micro computador anunciará para as caixas de som do ponto de ônibus apenas os ônibus que tiverem passando pela primeira antena e não tenham passado pela segunda e vice versa. Com a presença de duas antenas o sistema pode estimar a velocidade média no trecho e assim o tempo médio necessário para passar pelas duas antenas.

Veja a ilustração abaixo para melhor entender o funcionamento.

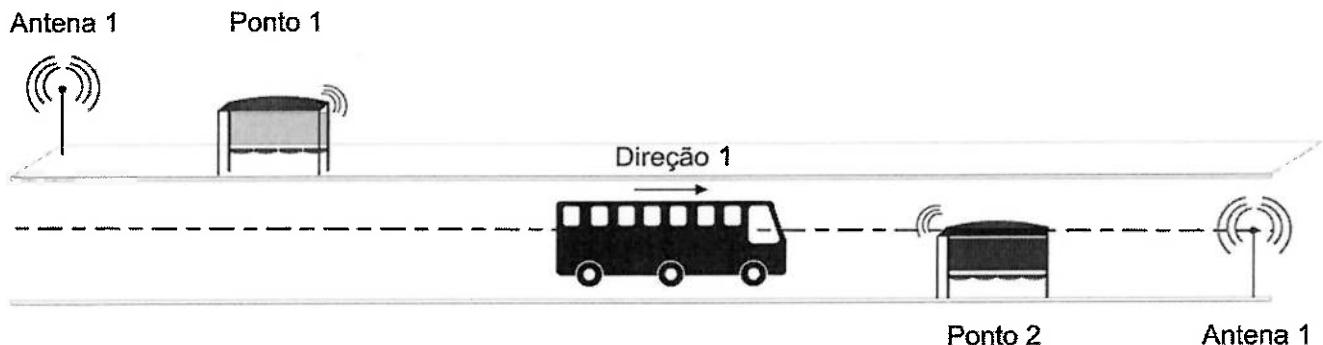


Figura 27 – Via de duas mão 1

No caso 1, o ônibus passará pela antena 1, assim o sistema detecta se este ônibus passou pela antena 2 dentro de um tempo médio gasto pelos ônibus no trecho, no caso negativo o sistema identifica que o ônibus está vindo na direção 1 e assim anuncia-o no ponto 2. Se o sistema detecta que o ônibus passou pelo antena 2 dentro do intervalo de tempo calculado o sistema ignora o sinal detectado pela antena.

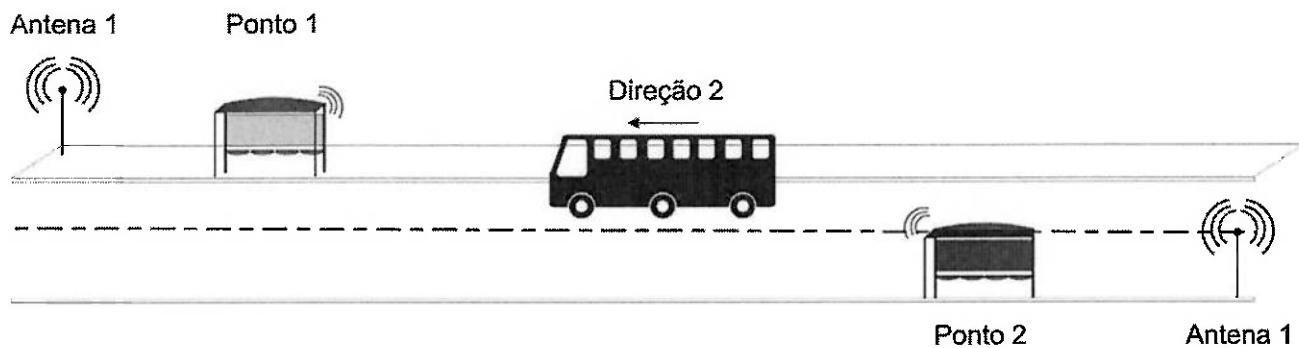


Figura 28 – Via de duas mãos 2

No caso 2, o ônibus passará pela antena 2, assim o sistema detecta se este ônibus passou pela antena 1 dentro de um tempo médio gasto pelos ônibus no trecho, no caso negativo o sistema identifica que o ônibus está vindo na direção 2 e assim anuncia-o no ponto 1. Se o sistema detecta que o ônibus passou pelo antena 1 dentro do intervalo de tempo calculado o sistema ignora o sinal detectado pela antena.

6.9 Ensaios Realizados

Este capítulo tem como objetivo apresentar os testes necessários para verificar o funcionamento do equipamento. Sendo assim pode-se encontrar no decorrer deste todos os ensaios que realizamos com cada sistema independentemente.

6.9.1 Ensaios com Regulador

Para os ensaios com o regulador de voltagem foi necessário montar o circuito e ir variando o capacitor a fim de se obter a voltagem e corrente nos padrões necessários para a alimentação dos circuitos, para realizar as medições foi utilizamos um multímetro digital convencional.

6.9.2 Ensaio com a porta serial

Para testar a porta serial inicialmente utilizamos um terminal USART ver figura 29, este terminal envia dados para a porta serial utilizando as normas de transmissão RS232, explicadas anteriormente neste texto.

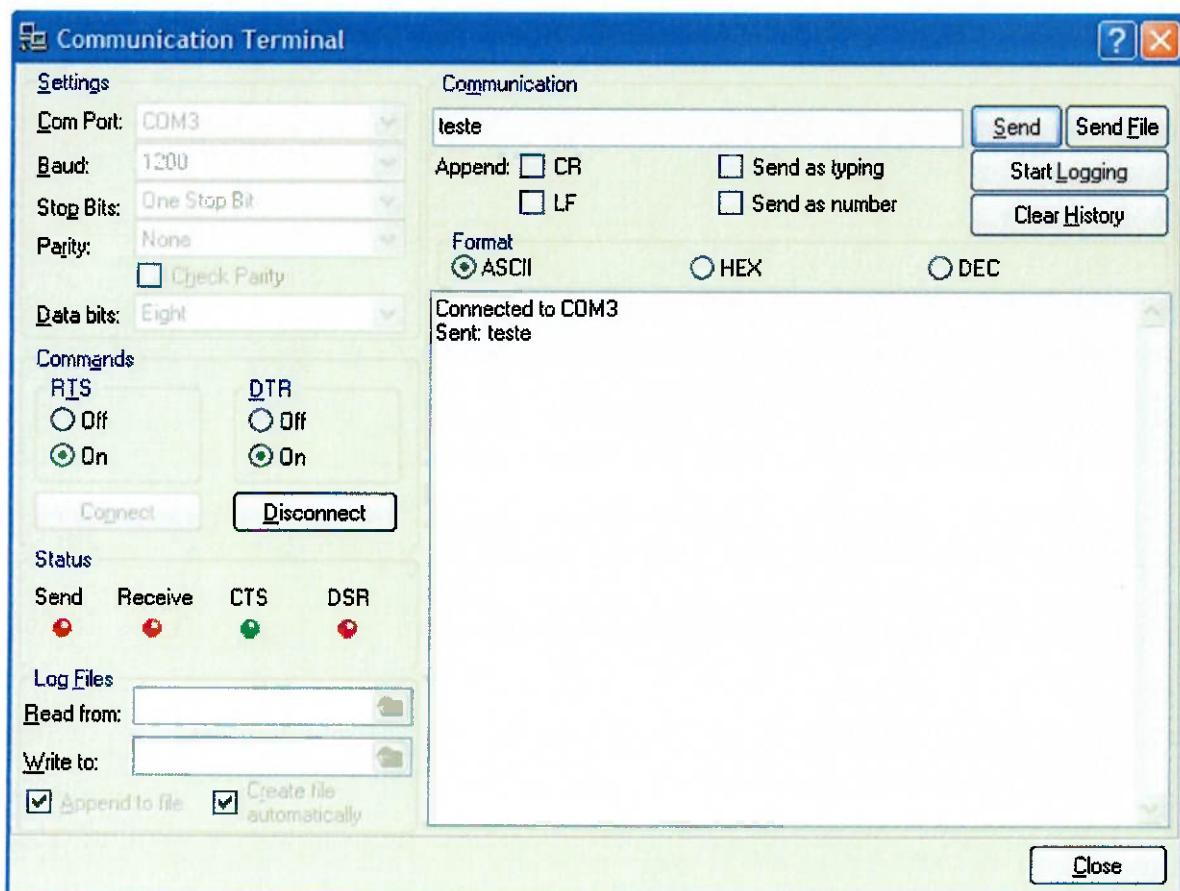


Figura 29 – Tela do terminal de comunicação serial

Com este terminal, inicialmente fizemos um curto-círcuito entre o RX (pino de recebimento de dados) e TX (pino de envio de dados). Assim, enviando dados para porta serial através do terminal, verificamos o funcionamento da porta serial do micro computador.

O segundo passo para os testes com a porta serial foi utilizar o circuito com MAX232 assim verificar o seu funcionamento. Para isso utilizando este mesmo terminal montamos o circuito MAX232 e fizemos um curto-círcuito entre os pinos 9 (R2 out) e 10 (T2 in). Assim, enviando dados para porta

serial pelo terminal foi possível testar se este circuito estava funcionando corretamente.

O terceiro passo foi necessário desenvolver um programa em C++ que fizesse o mesmo que o terminal, isto é conectasse com a porta serial e enviassem e recebessem dados. Assim poderíamos testar se os códigos desenvolvidos estavam funcionando.

O quarto passo foi realizar testes com PIC tanto utilizando o terminal como o programa desenvolvido, (veja melhores detalhes no capítulo 6.9.3) em Ensaios com PIC.

O quinto passo foi programar os protocolos de comunicação para realizamos ensaios com os módulos RF, (veja melhores detalhes no capítulo 6.9.4) em Ensaios com Módulo RF.

6.9.3 Ensaios com o PIC

Como o PIC era o componente que mais tinha variável (gravador, oscilador, portas, circuito, compilador) e apenas com todas funcionando teríamos os primeiros resultados, os ensaios com o PIC tiveram que ser mais detalhados.

Primeiramente fizemos ensaios com o gravador, para isso utilizamos um programa compilado que piscava *leds*. Descarregamos este no PIC e assim, verificamos o funcionamento do gravador e do circuito o qual o PIC estava ligado. Para gravar o PIC utilizamos o programa IC-Prog. Veja abaixo as configurações necessárias para que este programa funcione corretamente.

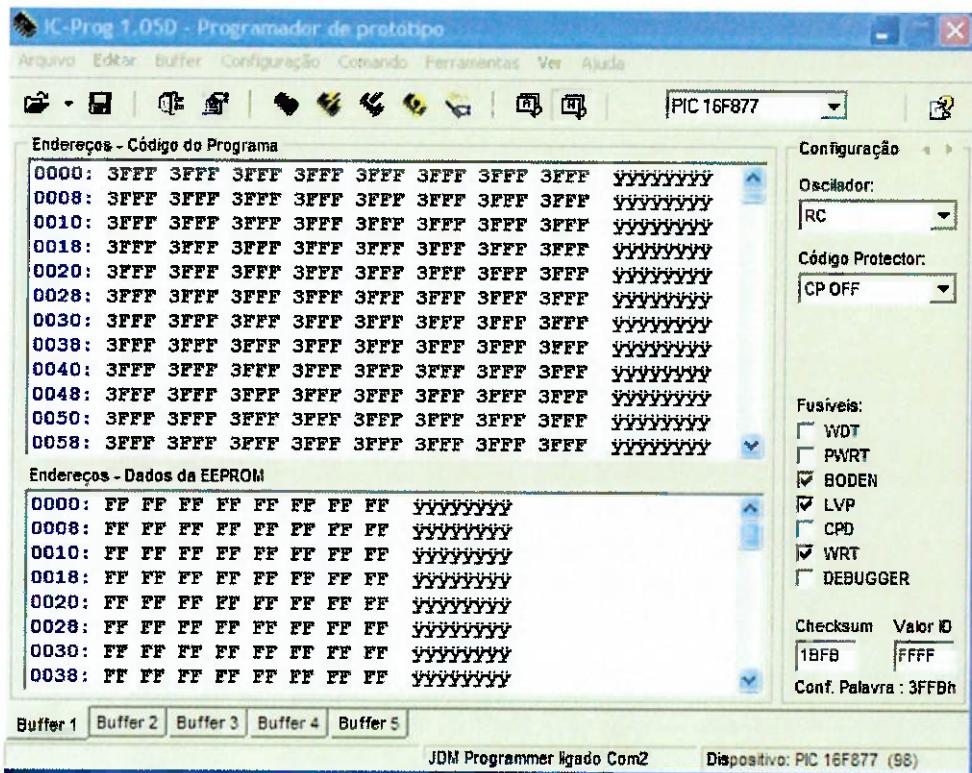


Figura 30 – Tela do gravador de PIC

Como segunda etapa, foi necessário testar o compilador. Para isso utilizamos o mesmo programa pisca LED só que não compilado, a escolha do compilador é muito importante, pois é necessário que este seja compatível com o gravador de PIC. O gravador testado que melhor funcionou o nosso gravador foi MICKO-C. Veja abaixo o código utilizado para que este teste fosse efetuado.

```
void main() {
    PORTC = 0;                      //Define porta C como zero
    TRISC = 0;                      //Define porta C como OUT

    while(1) {
        PORTC = ~PORTC;            //Inverte o valor na Porta C
        Delay_ms(1000);            //Aguarda 1s
    }
} //~!
```

Código 3 – Código fonte para piscar LED

A terceira etapa para seguirmos na fase de teste com PIC foi o teste da porta serial do PIC. Para isso fizemos a integração do PIC com o micro

computador. Utilizando a ligação representada na figura 31, o código do PIC representado no código 4 e o terminal fizemos testes de reflexão. Isto é, enviamos dados através da porta serial do computador para o PIC, o qual recebia os valores e devolvia para o computador e o terminal mostrava os valores recebidos.

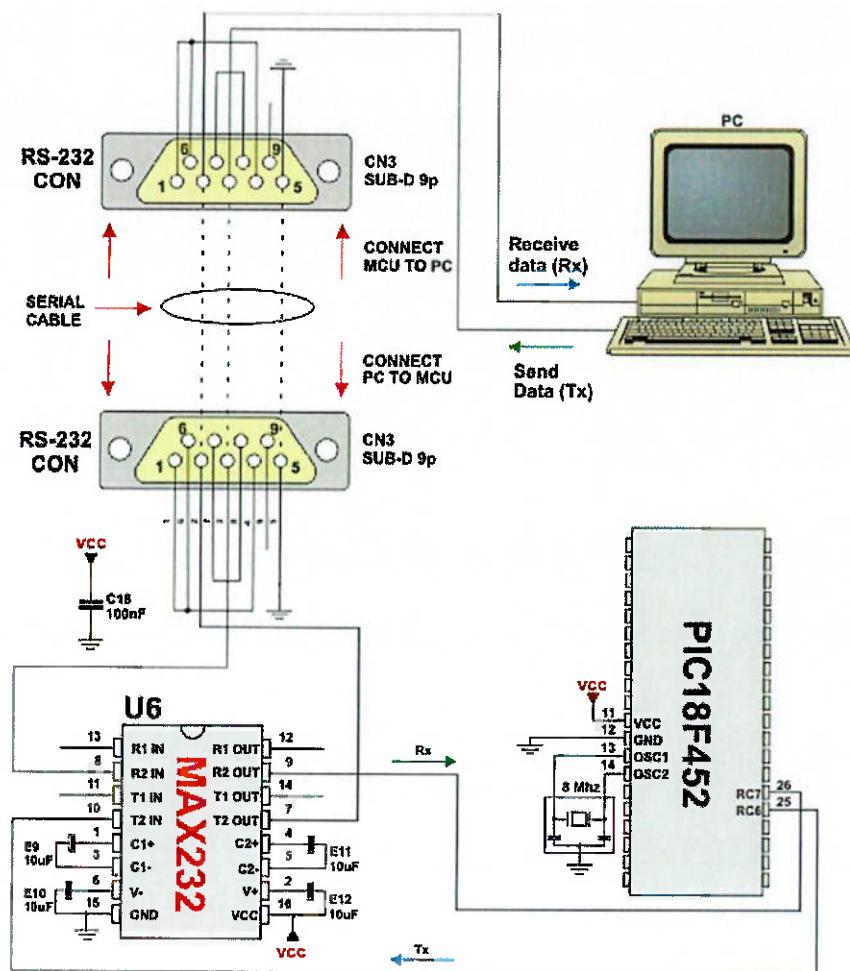


Figura 31 – Esquema de ligação para testar as portas seriais

```
unsigned short i;

void main() {
    USART_init(1200);                                // initialize USART module
                                                       // (8 bit, 2400 baud rate, no parity
    PORTB = 0;
    TRISB = 0;
    PORTB = ~ PORTB;
    while (1) {
        if (USART_Data_Ready()) {                    // if data is received
            i = USART_Read();                        // read the received data
            USART_Write(i);                          // send data via USART
        }
    }
} //~!
```

Código 4 – Código que testa a porta serial

O ultimo teste executado foi com a finalidade de verificar os botões. Assim foi feito um programa que acendia um LED e quando o botão fosse pressionado este se apagava. Veja abaixo um trecho do código que fazia este ciclo de operações.

```
void main(){
    char oldstate = 0;
    TRISB = 0xff;                                //Define porta B como in
    TRISD = 0;                                    //Define porta D como out

    do {
        if (Button(&PORTB, 1, 1, 1))             //se a porta B for acionado
            oldstate = 1;                         //altera a variavel
        if (oldstate && Button(&PORTB, 1, 1, 0)) { //se o botao foi solto
            PORTD = ~PORTD;                      //muda a porta D e assim o LED
            oldstate = 0;                         //muda a variavel
        }
    } while(1);                                //fim do LED
}
```

Código 5 – Código que testa os botões

6.9.4 Ensaios com o Módulo RF

Para verificar o funcionamento do módulo RF foi necessário primeiramente identificar se ambos estavam regulados para a mesma freqüência, para isto utilizamos um gerador de onda quadrada e um osciloscópio. Aplicamos uma onda quadrada no emissor e calibravos, rodando o parafuso, o receptor para a mesma freqüência do emissor.

Era necessário então testar o envio de dados através do módulo RF, para isso foi necessário utilizar o programa desenvolvido que contivesse os protocolos, e assim, utilizar o programa espelho do PIC para enviar dados e verificar os valores recebidos. Ao invés de colocar fios ligando o PIC ao PC colocamos o módulo RF.

6.9.5 Ensaios com o Banco de Dados

Para realizarmos a comunicação com o banco de dados como descrito anteriormente, utilizamos o programa DEVC++. O banco de dados utilizado, por sua vez foi o MySQL. Com a instalação dos *drivers* para possibilitar a conexão do software DEV-C++, o MySQL através do Windows pudemos iniciar ensaios na constituição do protótipo.

O programa deve realizar inclusão e leitura de dados do banco de dados de acordo com os critérios definidos pelo projeto. Desta forma, foi criada uma tabela (Onibus2) que contem três colunas como mostra a figura abaixo.

```
mysql> select * from onibus2;
+-----+-----+-----+
| serie | som  | ts   |
+-----+-----+-----+
| abc2  | tada.wav | 2006-11-13 01:18:27 |
| acc4  | tada.wav | 2006-11-12 16:12:27 |
| abd4  | start.wav | 2006-11-12 17:27:13 |
+-----+-----+-----+
3 rows in set <0.00 sec>

mysql>
```

Figura 32 – Banco de dados das linhas de ônibus

A coluna “serie” é responsável por guardar um número de referência da linha de ônibus, a coluna “som” por guardar o nome do arquivo de som que será executado e a coluna “ts” (*timestamp*) por gravar a data e hora de passagem do ônibus.

Os ensaios iniciais foram realizados através da impressão de dados de saída das funções utilizadas no código para que essas fosse checadas. Assim que uma rotina fosse incluída está era verificada pela realização de testes para garantir que cada parte do programa funcionaria.

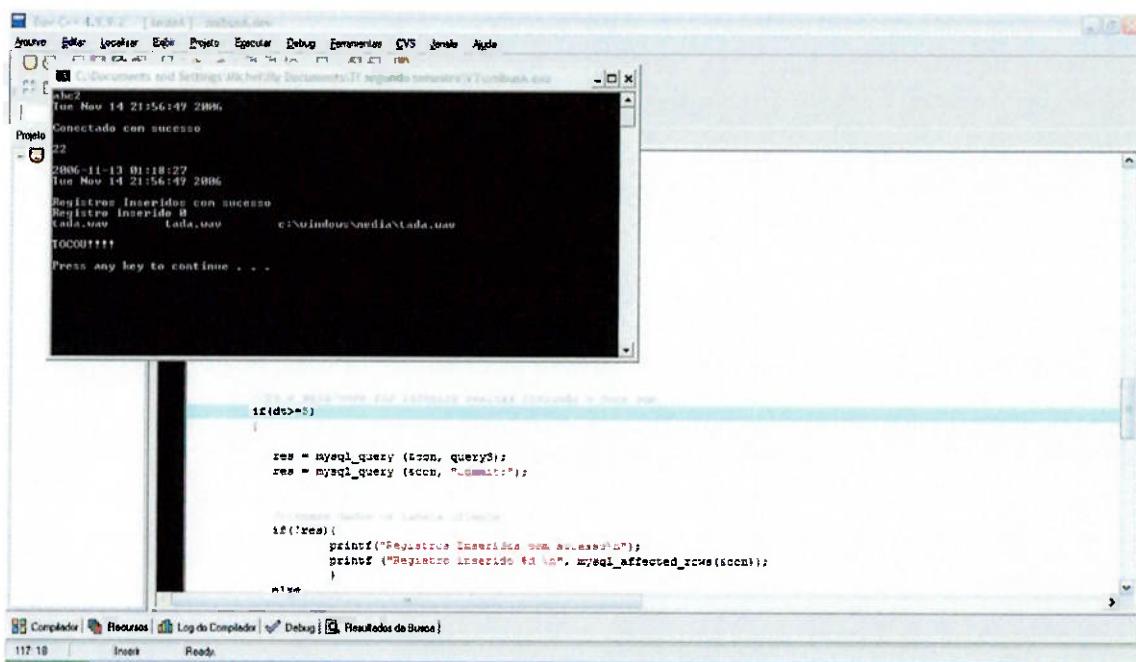


Figura 33 – Teste do programa

Simulando a entrada de um número de serie no formato que será recebido pela comunicação serial realizamos testes para verificar o funcionamento das funções de leitura e inclusão dos dados.

6.9.6 Ensaios com o Tocador de Som

O programa responsável por tocar o som é bastante simples. Utilizando a biblioteca windows.h do DEV-C++ realizamos a execução do arquivo indicado na função. Esta utiliza o programa padrão do Windows para a execução de arquivos de som. O teste desta se deu pela simples execução da função a partir de uma entrada que contem o nome do arquivo a ser executado no corpo do programa de banco de dados, de onde provem essas entradas. Para garantir que essa função é executada, também imprimimos um aviso para verificação.

6.10 Programação do PIC

Neste capítulo podem-se encontrar detalhes sobre o desenvolvimento do software que programou o PIC. Este programa tem como objetivo enviar o ID do ônibus através de uma conexão serial, e esta, por sua vez enviar dados através de um emissor de radiofreqüência.

6.10.1 Software de Programação

Ao iniciarmos este projeto percebemos que uma das dificuldades encontradas foi encontrar um software de compilação de C para Assembly que fosse de fácil utilização e tivesse uma integração com o gravador de PIC utilizado.

No inicio utilizávamos o software MPLAB, mas devido a dificuldades devido ao software não converter C em Assebly resolvemos alterar. Começamos a utilizar o PIC-C, não tivemos sucesso pois não foi compatível com nosso gravador.

Finalmente começamos a utilizar o software MIKRO-C, este de fácil utilização e completa compatibilidade com o gravador.

6.10.2 Etapas para o desenvolvimento

Antes de iniciarmos o desenvolvimento em si do programa que iria compor do micro processador PIC foram feitos diversos testes com este. Estes testes tinham como objetivo principal verificar se todos os dispositivos necessários para a utilização do PIC estavam funcionando e também se os nossos conhecimentos eram suficientes para operá-lo. Após feitos os testes, verificando assim o funcionamento do gravador, do circuito, dos pinos, da porta serial e do compilador (veja maiores detalhes no capítulo 8.9) foi iniciado a programação do método em a ser utilizado em si.

Inicialmente foi definido o baud rate em 1200 bps, isto é muito importante, pois o módulo RF não suporta velocidades mais rápidas do que esta, e assim temos que adaptar o PIC para esta velocidade, reduzindo o clock de freqüência de 20MHz para 10MHz.

Com a taxa de transmissão definida foi necessário definir se gostaríamos de LEDs para monitoramento e como estes indicariam um funcionamento regular. A fim de melhorar a velocidade de processamento decidimos que manteríamos os LEDs acessos quando o sistema estivesse funcionando corretamente, ficou decidido utilizar a porta D para estes LEDs.

Para o protótipo ficou decidido que cada PIC teria seis códigos diferentes e estes seriam definidos por acionamentos em três pinos, sendo assim possível fazer as seis combinações diferentes, ficou decidido em utilizar a porta B para este dispositivo.

Finalmente foi feita a integração de todas as funções e a montagem do protocolo, e assim, testado todo o dispositivo.

6.10.3 Descrição de Funcionamento

Inicialmente o programa inicia o PIC, definindo freqüência de utilização, e outras configurações necessárias para o funcionamento do mesmo, neste mesmo bloco do programa definiram-se as variáveis globais do programa que no caso é o número ID para as seis possibilidades de botões, o valor atual da posição dos botões e uma variável i, que recebe dados via porta serial caso algo seja enviado para o PIC em uma possível reprogramação.

Após esta primeira etapa o programa acende os LEDs, colocando a porta D e 5V. Assim o programa está preparado para entrar em um *looping* infinito durante toda a utilização deste dispositivo.

Dentro do *looping* infinito o programa lê os botões, caso exista alguma diferença com o valor gravado na memória o PIC verifica qual a nova combinação e assim constrói a string com os protocolos de comunicação. Ao final da string ele soma os bytes de gera o valor de SUM (para maiores detalhes ver capítulo 6.7), para mostrar que alguma alteração foi feita o programa pisca os Leds.

Com a string a ser enviada pronta o programa entra em um novo *looping* onde envia byte a byte através da porta serial para o emissor de RF.

Logo após o programa recomeça no *looping* infinito.

Veja abaixo um esquema de funcionamento do programa e logo após alguns trechos relevantes do código.

Rotina PIC

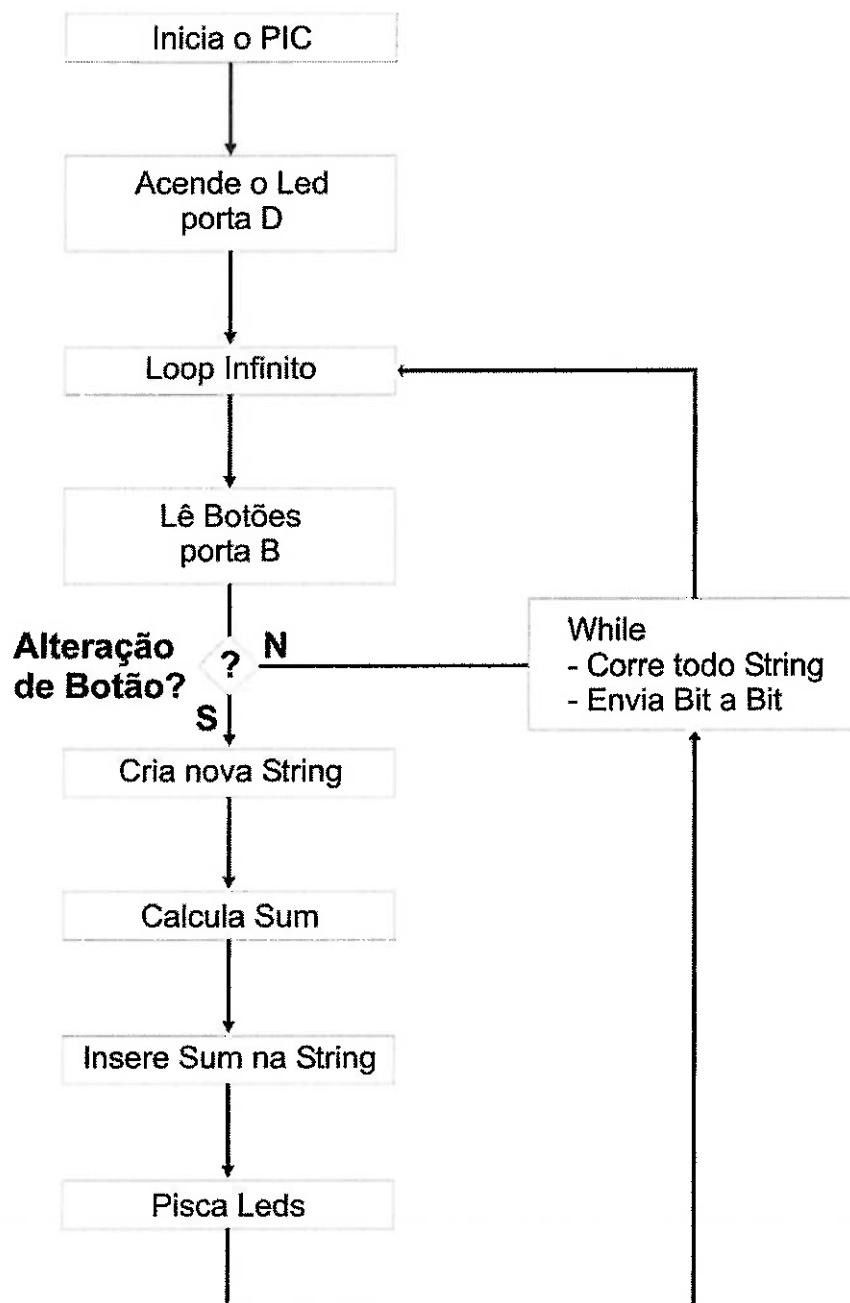


Figura 34 – Diagrama de blocos do programa do PIC

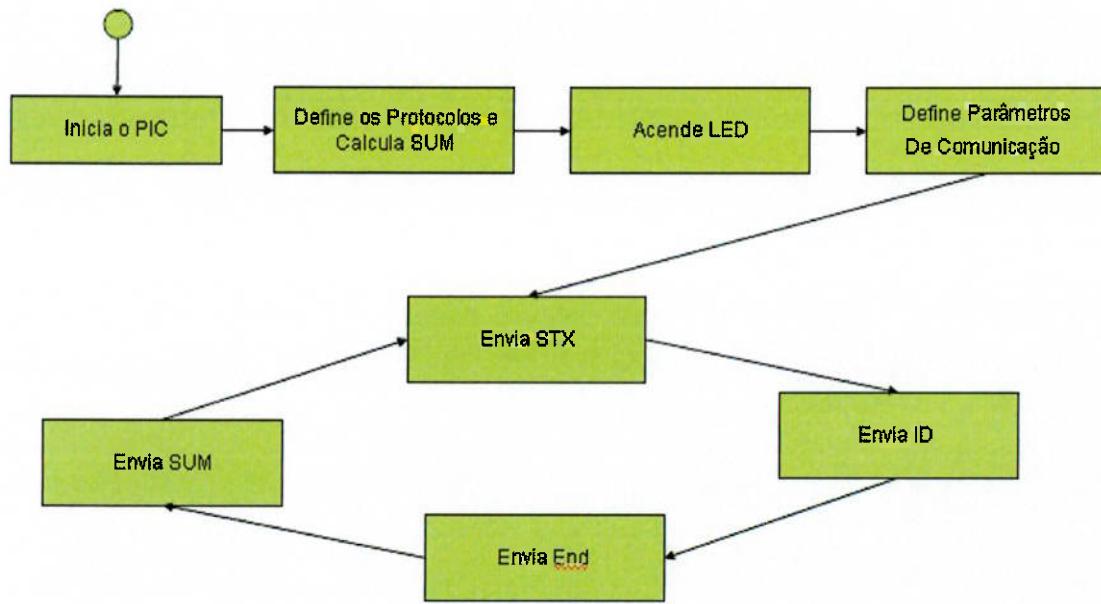


Figura 35 – Esquema do programa do PIC

O programa gravado no micro controlador PIC ocupou aproximadamente 18% da memória RAM. Veja abaixo algumas das principais funções do micro controlador.

Esta função tem como objetivo montar o ID do ônibus, o padrão definido foi no primeiro byte colocar a região do ponto de ônibus (Leste, Oeste, Norte, Sul, Centro), o segundo byte é para referenciar o nome do ponto final (no caso Butantã) e os outros dois bytes tem como objetivo dar um número para o ônibus 65535 números.

```

void monta_code() {
    //ID do primeiro ônibus
    code1[0] = '0';
    code1[1] = 'B';
    code1[2] = '0';
    code1[3] = '1';
}
    
```

Código 6 – Monta um ID de ônibus.

Esta função tem como objetivo detectar os botões selecionados, para isso ela lê através de uma função *Button* que elimina o *bouncing* e retorna verdadeiro ou falso. Podemos assim ter sete combinações possíveis.

```
void verifica_botao(){
    //caso Botao 1 = 0 e Botao 2 = 0 Botao 3 = 0
    if (!Button(&PORTB, 0, 1, 1) && !Button(&PORTB, 1, 1, 1) && !Button(&PORTB, 2, 1, 1))
        botao = 1;
    //caso Botao 1 = 1 e Botao 2 = 0 Botao 3 = 0
    if (Button(&PORTB, 0, 1, 1) && !Button(&PORTB, 1, 1, 1) && !Button(&PORTB, 2, 1, 1))
        botao = 2;
    //caso Botao 1 = 0 e Botao 2 = 1 Botao 3 = 0
    if (!Button(&PORTB, 0, 1, 1) && Button(&PORTB, 1, 1, 1) && !Button(&PORTB, 2, 1, 1))
        botao = 3;
    //caso Botao 1 = 1 e Botao 2 = 1 Botao 3 = 0
    if (Button(&PORTB, 0, 1, 1) && Button(&PORTB, 1, 1, 1) && !Button(&PORTB, 2, 1, 1))
        botao = 4;
    //caso Botao 1 = 0 e Botao 2 = 0 Botao 3 = 1
    if (!Button(&PORTB, 0, 1, 1) && !Button(&PORTB, 1, 1, 1) && Button(&PORTB, 2, 1, 1))
        botao = 5;
    //caso Botao 1 = 1 e Botao 2 = 0 Botao 3 = 1
    if (Button(&PORTB, 0, 1, 1) && !Button(&PORTB, 1, 1, 1) && Button(&PORTB, 2, 1, 1))
        botao = 6;
    //caso Botao 1 = 1 e Botao 2 = 1 Botao 3 = 1
    if (Button(&PORTB, 0, 1, 1) && Button(&PORTB, 1, 1, 1) && Button(&PORTB, 2, 1, 1))
        botao = 1;
}
```

Código 7 – Função que verifica a o botão apertado.

Esta função tem como objetivo montar a string final de envio pela porta serial do computador. Primeiramente ela coloca no primeiro byte o valor 0x02 que é o START byte, depois os próximos quatro bytes são dados de envio, o ID do ônibus, no quinto byte, a função coloca o END byte e finalmente finalizando a função o programa calcula a soma de todos os bytes e grava na posição seis da string. O código '0' tem o objetivo de finalizar a string.

```
//cria a string a ser enviada colocando os protocolos
cria_protocolo() {
    unsigned int cont;
    unsigned j;
    cont = 0;
    stringenviar[0] = 0x02;                                //protocolo Start
    stringenviar[1] = dados[0];                            //dados ID
    stringenviar[2] = dados[1];
    stringenviar[3] = dados[2];
    stringenviar[4] = dados[3];
    stringenviar[5] = 0x01;                                //protocolo END
    for(j = 1; j < 6; j++) {    //calcula SUM
        cont = cont + (int)(stringenviar[j]);
    }
    stringenviar[6] = (char)(cont);                         //SUM
    stringenviar[7] = '\0';
}
```

Código 8 – Função que cria e consolida a string de protocolos

O método abaixo é o coração do programa, esta função ficará corrente interrompontamente até que o PIC seja desligado. O objetivo principal desta função é enviar os bytes, uma a um, através da porta serial.

Para fazer o envio é utilizada a função USTAR_Write que é uma biblioteca do compilador em C.

```
while (1) {
    //colocar o FOR para mandar todos os dados
    for(k = 0;k < 8; k++){
        USART_Write(stringenviar[k]);                  // send data via USART
    }
}
} //~!
```

Código 9 – Método de envio da porta serial

6.11 Programação do Micro Computador

Este capítulo tem como objetivo explicar os códigos implementados no micro computador. Para melhor entendimento e organização, este foi divido em subitens os quais explicam detalhadamente os três programas independentemente, o leitor na porta serial, o gerenciador do banco de dados e o tocador de som.

6.11.1 Esquema de integração

Primeiramente falaremos como que estes programas são integrados.

O leitor da porta serial quando recebe um dado que respeita os protocolos de comunicação, envia uma string com o ID recebido para o programa leitor de banco de dados.

Este por sua vez, faz um SELECT no banco e verifica se este ID é cadastrado no banco, em caso afirmativo o programa verifica as rotinas de logísticas (ver detalhes no capítulo 6.8) e captura do banco de dados um string com o endereço do arquivo que contem o som a ser tocado.

O tocador de som recebe esta string e roda o arquivo de som, emitindo assim o som desejado.

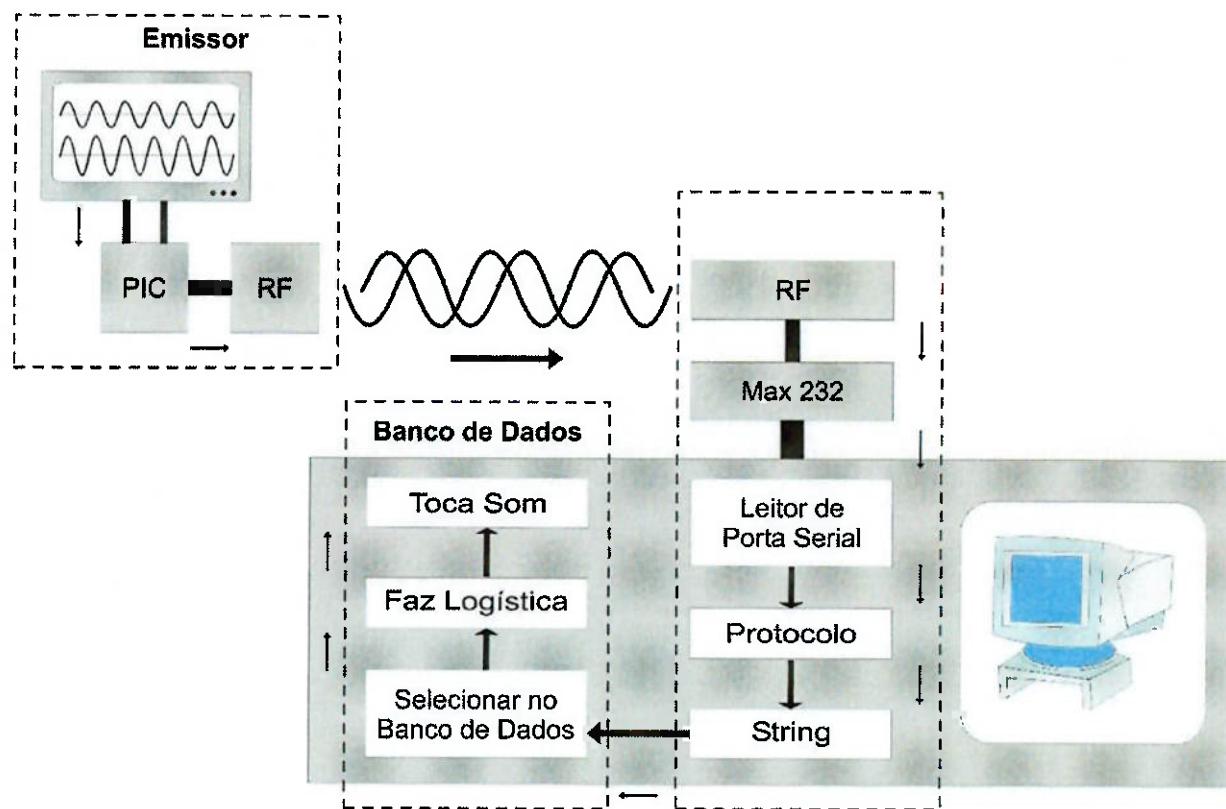


Figura 36 – Diagrama do sistema de integração

6.11.2 Leitor da porta serial

Neste capítulo encontra-se detalhes sobre o programa que conecta à porta serial, lê os arquivos, interpreta os protocolos a string lida via comunicação RF.

6.11.2.1 Funcionamento

Este programa usa com base funções API do Windows para fazer a abertura e a leitura de dados da porta serial.

O sistema primeiramente define os parâmetros para a conexão com a porta serial, neste caso utilizamos a porta COM2, com uma taxa de transferência de 1200bps, com transferência de 8bits sem paridade.

Com a porta serial aberta o programa define os protocolos de comunicação a serem respeitados. (ver detalhes no capítulo 6.7).

Com todos os parâmetros definidos o programa entra em um *looping* infinito que para cada repetição ele primeiramente verifica se tem algum dado novo lido pela porta serial. No caso positivo, ele pega este dado e verifica se este é o START byte para o envio de dados. O programa faz esta operação até que este byte seja encontrado. Quando isto for satisfeito, o programa grava em uma memória os próximos dois bytes de dados, referentes ao ID no veículo.

Logo após o programa lê o próximo dado da porta serial e verifica se é o END byte. No caso negativo o programa reinicia estas etapas e começa a procurar o START byte novamente. No caso positivo o programa recebe mais um byte, SUM, verifica se este é realmente o valor esperado. No caso negativo o programa reinicia estas etapas e começa a procurar o START byte novamente. No caso positivo o programa pega os dois dados, ID, já gravados e envia para o programa que gerencia o banco de dados.

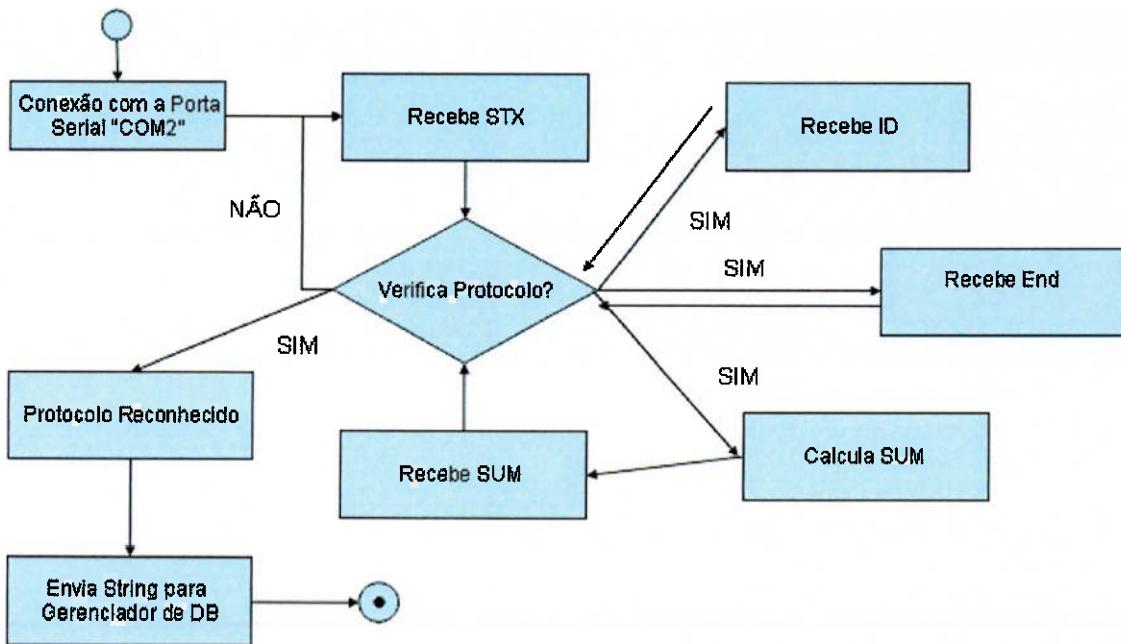


Figura 37 – Esquema do software leitor de porta serial

Rotina Leitor Porta Serial

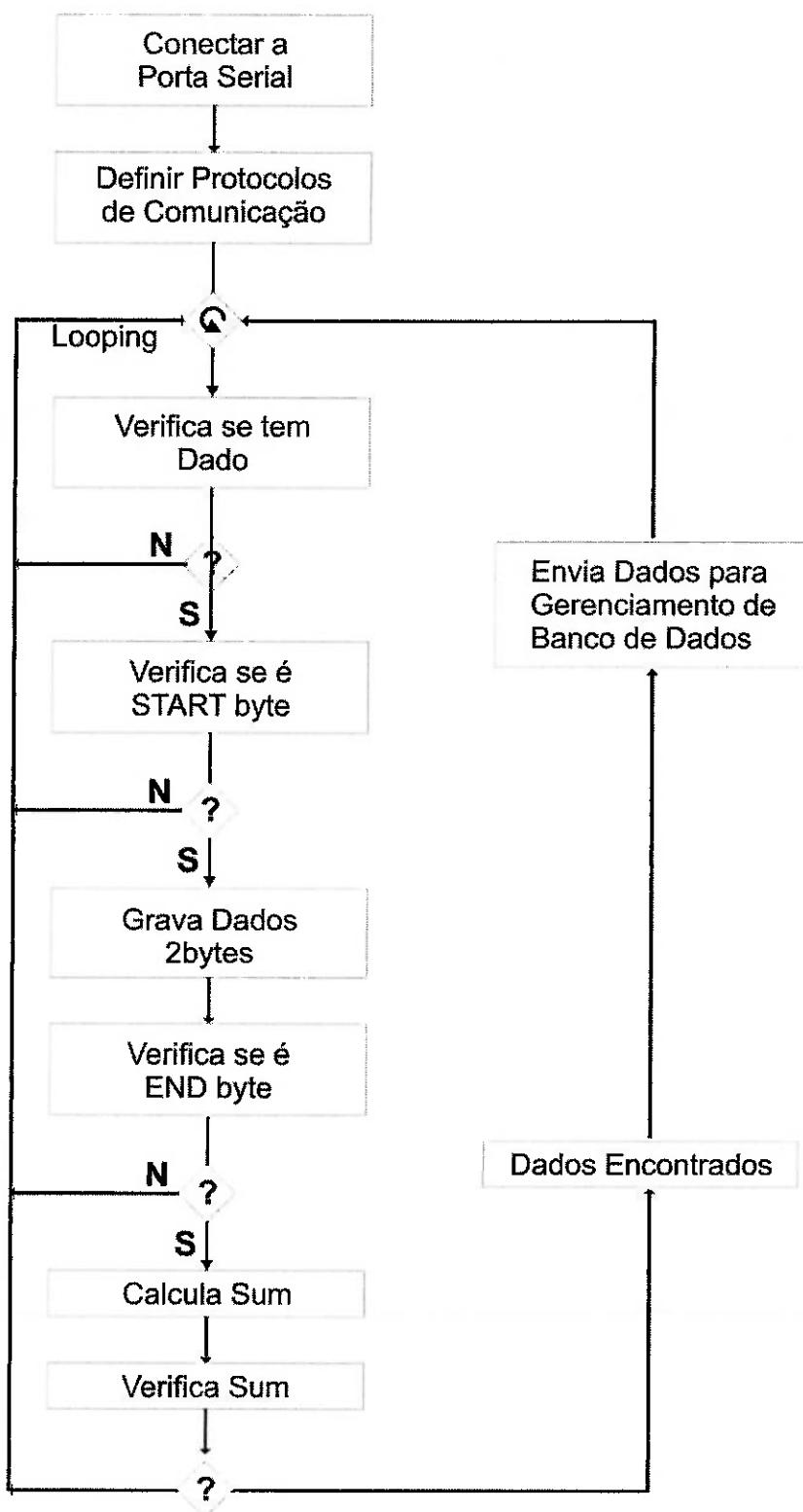


Figura 38 – Diagrama de blocos do leitor de porta serial

6.11.2.2 Etapas de desenvolvimento

Inicialmente foram feitos testes para verificar se a conexão com porta serial estava funcionando, para isso, foi feito um curto circuito entre o RX e o TX (ver detalhes no capítulo 6.9.2). Com esta etapa finalizada iniciou-se o desenvolvimento de uma rotina que gerava e verificava os protocolos comunicação, para testar estas rotinas, digitamos entradas de dados e assim, verificamos as saídas.

Com as rotinas de comunicação com a porta serial e as rotinas de interpretação de Strings funcionando foi feita a integração das duas partes e assim constituído o programa.

6.11.2.3 Funções e códigos utilizados

O programa basicamente contém uma função que verifica os protocolos de comunicação, uma função que conecta com a porta serial e uma função main() que contém um *looping* infinito e fica chamando as funções mencionadas.

Para as funções de protocolo veja detalhes no capítulo 6.8 e as funções da API do Windows podem ser entendidas melhor no próximo capítulo.

6.11.2.3.1 Funções da API do Windows utilizadas

Este capítulo tem como objetivo descrever as funções API do Windows utilizadas para estabelecer, enviar e receber dados via conexão serial do micro computador. Veja abaixo uma breve descrição sobre as principais funções utilizadas.

- **CreateFile():** Utilizada para abrir a Porta Serial.
- **CloseHandle():** Utilizada para fechar a Porta Serial.
- **ReadFile() e WriteFile() :** Utilizadas para lê e enviar dados através da Porta Serial.

- **GetCommState()** e **SetCommState()**: Utilizadas para obter e alterar as configurações da Porta Serial, como velocidade, bit de dados, paridade etc.
- **BuildCommDCB()**: Utilizada para preencher os campos da estrutura DCB com valores especificados numa string de controle de dispositivos.
- **GetCommTimeouts()** e **SetCommTimeouts()** : Utilizadas para obter e alterar os parâmetros atuais de Timeouts associados às funções WriteFile() e ReadFile();

Veja abaixo maiores detalhes sobre as funções utilizadas no programa de leitura da porta serial.

```
HANDLE hCom; // Handle para a Porta Serial (identificador).
char *NomePorta = "COM1"; //COM1, COM2...COM9 ou portas virtuais "\.\COMx".
```

```
hCom = CreateFile(
    NomePorta, //Nome da porta.
    GENERIC_READ | GENERIC_WRITE, //Para leitura e escrita.
    0, //Zero) Nenhuma outra abertura será permitida.
    NULL, //Atributos de segurança. (NULL) padrão.
    OPEN_EXISTING, //Criação ou abertura.
    0, //Entrada e saída sem overlapped.
    NULL //Atributos e Flags. Deve ser NULL para COM.
);
```

```
if(hCom == INVALID_HANDLE_VALUE)
    return false; //Erro ao tentar abrir a porta especificada.
```

Código 10 - Uso da função CreateFile() da API do Windows

A função CreateFile() da API do windows é responsável pela abertura da Porta Serial. Ela aceita vários parâmetros, sendo que o primeiro é o nome da porta "COM1", "COM2", etc. O segundo especifica o tipo de acesso: leitura (GENERIC_READ), escrita (GENERIC_WRITE) ou leitura e escrita (GENERIC_READ | GENERIC_WRITE), o caracter "|" representa o

operador (OR) e sua função no Exemplo 1 é combinar as duas constantes formando um novo valor. O terceiro parâmetro especifica o compartilhamento, se for definido como 0 (zero), a Porta não será compartilhada, ou seja, enquanto seu programa estiver usando a Porta Serial nenhum outro poderá abri-la. Os outros parâmetros serão usados como padrão.

hCom é uma variável do tipo HANDLE (identificador) que armazenará todas as informações necessárias sobre o dispositivo aberto, no nosso caso, a Porta Serial. Essa variável será utilizada como parâmetro para as funções que irão configurar, alterar, lê e escrever a Porta Serial.

Se houver algum tipo de erro na abertura da Porta Serial, a função CreateFile() retornará um valor do tipo INVALID_HANDLE_VALUE para a variável hCom. Portanto, é imprescindível que o programa teste a variável hCom para checar se houve erro, antes de usar as demais funções de acesso a Porta Serial.

DCB **dcb**; //Estrutura DCB é utilizada para definir todos os parâmetros da comunicação.

```
if( !GetCommState(hCom, &dcb)  
    return false; ///// Erro na leitura de DCB.  
  
dcb.BaudRate = CBR_19200;  
dcb.ByteSize = 8;  
dcb.Parity = NOPARITY;  
dcb.StopBits = ONESTOPBIT;  
  
//Define novo estado.  
if( SetCommState(hCom, &dcb) == 0 )  
    return false; //Erro.
```

Código 11 - Uso das funções GetCommState() e SetCommState()

As funções GetCommState() e SetCommState() trabalham juntas e têm como objetivos obter as configurações atuais da Porta Serial depositadas na estrutura DCB, como também alterar seus atributos.

Para mudar as configurações atuais da Porta Serial pode-se usar o código do Código 11, ou mesmo o código do Código 12, sendo este mais simples.

DCB dcb; //Estrutura DCB é utilizada para definir todos os parâmetros da comunicação.

```
if( !GetCommState(hCom, &dcb))
    return false; /// Erro na leitura de DCB.
```

```
BuildCommDCB("19200,N,8,1", &dcb); //Atribui os valores a estrutura dcb.
```

//Define novo estado.

```
if( SetCommState(hCom, &dcb) == 0 )
    return false; //Erro.
```

Código 12 - Uso das funções GetCommState(), SetCommState() e BuildCommDCB()

No exemplo acima, a função BuildCommDCB() atribuir a string "19200,N,8,1" a membros específicos da estrutura DCB.

Tabela 8 - Constantes e valores da API do Windows (TM) utilizadas para configurar a Porta Serial

BaudRate <i>Velocidade</i>	ByteSize <i>Tamanho dos dados</i>	Parity <i>Paridade</i>	StopBits <i>Bits de parada</i>
CBR_110	CBR_38400	5	EVENPARITY (par)
CBR_300	CBR_56000	6	MARKPARITY (marca)
CBR_600	CBR_57600	7	NOPARITY (nenhuma)
CBR_1200	CBR_115200	8	ODDPARITY (ímpar)
CBR_2400	CBR_128000		
CBR_4800	CBR_256000		
CBR_9600			
CBR_14400			
CBR_19200			

```
COMMTIMEOUTS CommTimeouts;

if( GetCommTimeouts(hCom, &CommTimeouts) == 0 )
    return false; //Erro.

//Define novos valores.
CommTimeouts.ReadIntervalTimeout = 2;
CommTimeouts.ReadTotalTimeoutMultiplier = 0;
CommTimeouts.ReadTotalTimeoutConstant = 2;
CommTimeouts.WriteTotalTimeoutMultiplier = 5;
CommTimeouts.WriteTotalTimeoutConstant = 5;

if( SetCommTimeouts(hCom, &CommTimeouts) == 0 )
    return false; //Erro.
```

Código 13 - Uso da função GetCommTimeouts() e SetCommTimeouts()

As funções GetCommTimeouts() e SetCommTimeouts() trabalham juntas e têm como objetivos obter e modificar as configurações atuais dos *timeouts* da Porta Serial. Quaisquer mudanças nos valores dos atributos da estrutura COMMTIMEOUTS afetarão consideravelmente os tempos de leitura e escrita na Porta Serial. Os valores de tempos definidos no Exemplo 4 foram os mais adequados para ler a placa controladora de acessos. Outros valores podem ser testados.

```
#define LEN_BUFFER 100; //Define o tamanho do buffer.

DWORD BytesLidos = 0;
char BufferRecebe[LEN_BUFFER]; //Para armazenar a string a ser lida.

ReadFile( hCom, BufferRecebe, LEN_BUFFER, &BytesLidos, NULL );
```

Código 14 – Uso da função ReadFile()

É através da função ReadFile() que é feita a leitura dos dados que chegam na Porta Serial. Esta função aceita 5 parâmetros, sendo o primeiro (hCom) o identificador da Porta Serial retornado na abertura da mesma

através da função CreateFile(). O segundo é um buffer, ou seja, é um local temporário na memória onde os dados lidos da Porta Serial serão armazenados. O terceito informa à função ReadFile(), qual o tamanho do buffer de dados (BufferRecebe). O quarto é uma variável do tipo ponteiro inteiro longo, onde a função ReadFile() armazenará a quantidade exata de bytes lidos. O quinto e último é um ponteiro para uma estrutura "overlapped", no nosso caso não será utilizada, portanto definimos este parâmetro com (NULL).

A cada chamada à função ReadFile(), os dados lidos da Porta Serial são armazenados na variável BufferRecebe, e a quantidade total de bytes são armazenados na variável BytesLidos.

```
#define LEN_BUFFER 100; //Define o tamanho do buffer.

DWORD BytesEscritos = 0;
int TamaString; //Para armazenar o tamanho da String.
char BufferEnvia[LEN_BUFFER]; //Para armazenar a string a ser enviada.
strcpy(BufferEnvia, "CMD05#1#025\r\n"); //Prepara a string a ser enviada.
TamaString = strlen(BufferEnvia); //Calcula o tamanho da string a ser enviada.

WriteFile( hCom, BufferEnvia, TamaString, &BytesEscritos, NULL );
```

Código 15 – Uso da função WriteFile()

É através da função WriteFile() que nossos programas enviam dados pela Porta Serial. Esta função aceita 5 parâmetros, sendo o primeiro (hCom) o identificador da Porta Serial retornado na abertura da mesma através da função CreateFile(). O segundo é um buffer, ou seja, é um local na memória onde deverá estar depositados os dados a serem enviados para a Porta Serial. O terceiro informa à função WriteFile() qual o tamanho do buffer de dados (BufferEnvia). O quarto é uma variável do tipo ponteiro inteiro longo, onde a função WriteFile() armazena a quantidade exata de bytes escritos. O quinto e último é um ponteiro para uma estrutura "overlapped", no nosso caso não será utilizada, portanto definimos este parâmetro com (NULL).

A cada chamada à função WriteFile(), os dados armazenados no buffer (BufferEnvia) serão enviados para Porta Serial.

```
CloseHandle( hCom ); //Fecha a porta aberta anteriormente por  
CreateFile().
```

Código 16 – Uso da função CloseHandle()

A função CloseHandle() fecha a Porta Serial aberta pela função CreateFile(), disponibilizando-a para que outros programas possam acessá-la normalmente. Obs.: As funções ReadFile() e WriteFile() são ser chamadas a partir de *Threads* (linhas de execução ou processos que rodam paralelo ao programa). Assim obtém-se uma boa performance de programas que trabalham com a Porta Serial.

6.11.3 Comunicação com banco de dados

6.11.3.1 Descrição do código

O código desenvolvido em linguagem C++ é responsável pela comunicação com o banco de dados. Ele realiza a consulta às informações da linha de ônibus dado uma entrada contendo o número de serie de um modulo RF associado ao ônibus.

O código se utiliza da biblioteca mysql.h que nos fornece funções de acesso ao banco de dados. Através dessas realizamos as operações necessárias. A variável “con” é uma variável dinâmica que realiza as operações entre o código em C++ e o banco de dados.

O código recebe uma string contendo quatro caracteres char que servem como referência para a identificação da linha de ônibus. Esse valor deve ser encontrado no banco de dados, retornando a data e hora do ultimo ônibus de mesmo número que foi incluído no banco de dados. Encontrado esse valor, o programa realiza um teste condicional para verificar o tempo da ultima passagem e caso esse teste seja validado, realiza-se a inclusão da data atual no banco de dados seguido da execução do arquivo de som através da extração dos dados contidos na tabela.

Inicialmente, através da variável “com” o programa faz a conexão com o MYSQL, fornecendo a senha, nome de usuário e o *host*. Caso ocorra erro na realização das operações com o banco de dados esses são descritos na condição de *else*.

A rotina então realiza um *query*, através da mesma variável “con” responsável por devolver o valor da data/hora contida na tabela. O *query* é composto pela concatenação da string recebido pela função e as palavras necessárias ao comando no MYSQL.

O valor recebido em uma string é comprado com o valor de data hora atual retornadas pela função de tempo consultada. Essa é a comparação necessária para garantir que se execute apenas uma vez o som proveniente do sistema. Se essa operação for validada o programa realiza, também através da variável “com”, a inclusão da data/hora atual em que esse registro ocorre e indica que o processo foi realizado. Após essa operação o programa realiza um outro *query* que devolve em uma string o nome do arquivo de som de deve ser executado. Esse valor serve de entrada para a função “toca”, que executa o arquivo. Assim, o programa é finalizado limpando as variáveis utilizadas, para a próxima utilização.

6.11.3.2 Bibliotecas utilizadas

`#include <cstdlib> e #include <iostream>` - Bibliotecas padrão do DEV-C++ para a inclusão de comandos em C++.

`#include <windows.h>` - Biblioteca do windows para a execução dos arquivos de som.

`#include <mysql\mysql.h>` - Biblioteca do DEV-C++ para a realização de operações com o MYSQL

`#include <time.h>` - Biblioteca para implementação do código de consulta a data/hora.

6.11.3.3 Funções utilizadas

No desenvolvimento dessa função foram utilizadas funções das bibliotecas necessárias para a nossa aplicação. As mais importantes para nosso uso são:

`strcat(serie2, serie);` - Concatena as *strings* para formação dos *querys*
`time (&rawtime);` - Cria a variável dinâmica para acesso ao tempo.
`localtime (&rawtime);` – Retorna a data/hora atual do computador.
`mysql_init(&con);` - Inicializa a variável “con” para a comunicação entre os programas.

`mysql_real_connect(&con, "host", "usuario", "senha", "basededados", 0, NULL, 0);` – Realiza a conexão com o banco através dos parâmetros do MYSQL.

`mysql_query(&con,query);` – Realiza o *query* na MYSQL através da variável “com”.

`mysql_error(&con);` – Apresenta o erro que ocorreu na operação no MYSQL.

`mysql_store_result(&con);` - Guarda o resultado da *query* realizada.

`mysql_fetch_row(resp)` – Retorna o valor da consulta do *query* em uma linha.

`mysql_free_result(resp);` - Limpa a variável de consulta.

`mysql_close(&con);` - Fecha a conexão com o MYSQL.

6.11.3.4 Etapas de desenvolvimento

Inicialmente através de uma pesquisa procuramos nos familiarizar com as características do sistema a ser montado. Para isso precisamos saber quais eram os componentes, procedimentos e comandos necessários a inclusão dessa estrutura.

Primeiramente realizamos a instalação dos programas DEV-C++ e MYSQL necessários para o desenvolvimento desta parte do projeto. Para que estes dois softwares possam realizar a conexão entre si são necessários alguns procedimentos. Instalamos o driver “libmysql-5.0.5-1sid.DevPak” e copiamos o arquivo mysql.dll na pasta windows/system32 do

sistema operacional Windows. Isso é necessário para que a comunicação seja feita a partir do sistema operacional através do API para Windows. Então, copiamos a biblioteca libmysql.a para a pasta dev-cpp/lib, para que possamos utilizar as funções disponíveis a biblioteca "mysql.h". No software de programação criamos um novo projeto no qual devemos escrever "-lmysql" no campo "linker" para incluirmos a biblioteca descrita. Esses arquivos foram coletados da internet.

Com isso passamos a nos dedicar com a aplicação em si. Neste contexto, após a análise de quais informações que seriam necessárias no banco de dados para o funcionamento do projeto passamos a criar a tabela. Esta serve para que possamos escrever o código em C++. Ela contém três colunas com informações quanto ao número de serie, o nome do arquivo de som e a data/hora atribuída ao evento da passagem do ônibus.

As linhas no MYSQL para essa tarefa são:

```
create table onibus2(serie char(5), som char(50), ts timestamp);
```

Assim, também inserimos valores na tabela para que possamos testar o código a ser elaborado através da linha:

```
insert into onibus2 values ('abc2','tada.wav', now());
```

Finalizada esta etapa iniciamos a implementação dos comandos no DEV-C++ para realizar as operações com a base de dados. A primeira coisa a se fazer são os comandos responsáveis pela conexão. Então escrevemos as rotinas de leitura e inclusão de informações no banco de dados. Esses passos são seguidos de testes para verificação do devido funcionamento do programa. Assegurado o funcionamento podemos definir a estrutura do sistema como descrito acima através da inclusão das rotinas que constituem o mesmo.

6.11.3.5 Esquema

A rotina de conexão com o banco de dados pode ser representada pelo esquema abaixo para facilitar a visualização do funcionamento do sistema.

Gerenciador de Banco de Dados

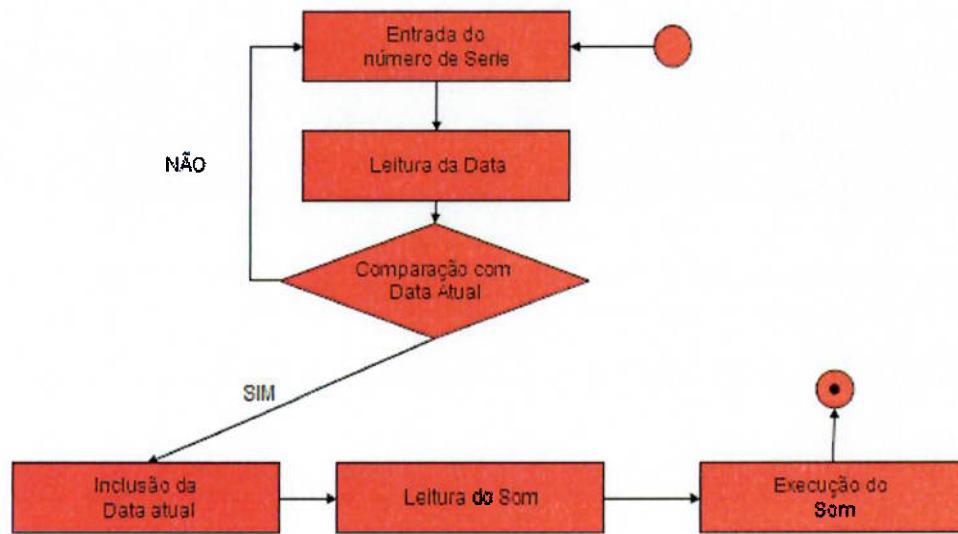


Figura 39 – Esquema do banco de dados

6.11.3.6 Código fonte das principais partes

```
//Se a data/hora for inferior ao valor definido realize inclusão e toca som
if(dt>=5)
{
    res = mysql_query (&con, query3);
    res = mysql_query (&con, "commit");

    //insere dados na tabela
    if(!res)
    {
        printf("Registros Inseridos com sucesso\n");
        printf ("Registro inserido %d \n", mysql_affected_rows(&con));
    }
    else
        printf("ERRC na inserção %d:%s \n", mysql_errno(&con), mysql_error(&con));
}

//Consulta dados de som
if (mysql_query(&con,query))
{
    printf("Erro: %s \n",mysql_error(&con));
}

else
{
    resp = mysql_store_result(&con); //recebe a consulta
```

Código 17 – Código fonte para acesso ao banco de dados

6.11.4 Tocar SOM

6.11.4.1 Descrição do código

Esta função é bastante simples. Trata-se de uma função contida na biblioteca “windows.h” que executa o som encontrado no diretório indicado.

A função recebe o nome do arquivo de som para que este seja executado. A string recebida pela função é proveniente do banco de dados e essa é então concatenada com uma string contendo o diretório onde se encontra este arquivo.

6.11.4.2 Código fonte das principais partes

```
using namespace std;

int toca (char a[]){
    char destino[40] = "c:\\windows\\\\media\\\\";
    strcat(destino,a);
    printf("%s \n\n", destino);
    ShellExecute(NULL,NULL,destino,NULL,NULL,0);
    printf("TOCCOU!!!!\n\n");

    return (0);
}
```

Código 18 – Código fonte para tocar o som

7 Resultados e discussões

Os estudos efetuados durante todo o decorrer do projeto resultaram na elaboração de um protótipo.

A construção desse teve como base os módulos que foram pré-testados restando, assim, apenas a integração final para verificar se estes conseguiam se comunicar de forma eficiente. Devido a uma estruturação devidamente elaborada dos softwares desenvolvidos, a integração de todo o sistema foi feita com êxito, sem maiores dificuldades.

O protótipo final, como descrito anteriormente, é constituído por um emissor e um receptor.

O emissor (ver figura 40) é o componente fixado no interior do veículo, ele é responsável por emitir as ondas de rádio. O preço deste protótipo, item de importância em nosso projeto, foi R\$51,60. Este valor, contudo, pode ser expressivamente reduzido com a alteração do microcontrolador, passando de um dispositivo avançado para elaboração de protótipo para um simples de oito pinos, que pode ser gravado apenas uma vez. Assim, o custo seria reduzido para cerca de R\$20,00 reais. Além disso, com a fabricação em série deste sistema, tendo em vista que esses equipamentos têm alto valor agregado e, assim, devem ter um *markup* de aproximadamente 35%, acreditamos que o preço final do emissor será de próximo R\$23,50.

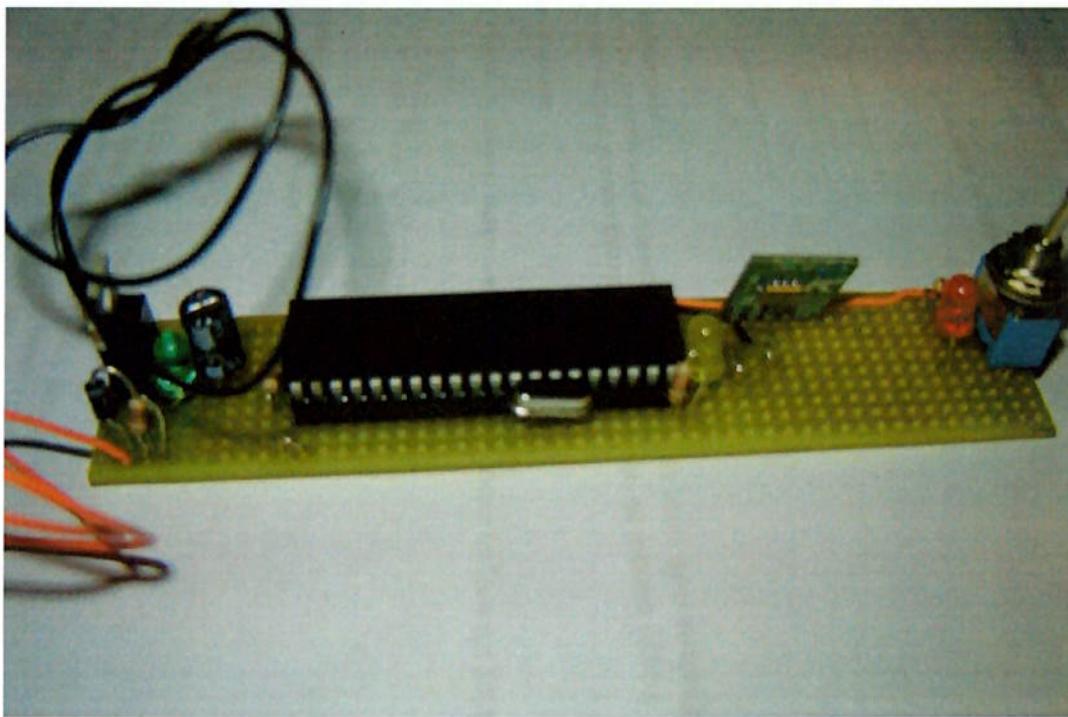


Figura 40 – Foto do emissor

O receptor (ver figura 41) é o equipamento fixado no ponto de ônibus, ele é responsável por receber os sinais de radiofreqüência e anunciar a linha em caixa de som.

O protótipo deverá ser acoplado a um microcontrolador da National com suporte para Windows [NATIONAL] que emula as funções efetuadas pelo micro computador. Pelos mesmos motivos do emissor, estimamos que o preço de fabricação em série deve ser 35% menor, assim o módulo receptor teria um custo próximo de R\$21,00. Devemos frisar que também é necessário para cada ponto de ônibus uma placa da National que custa aproximadamente US\$150,00 e uma caixa de som.

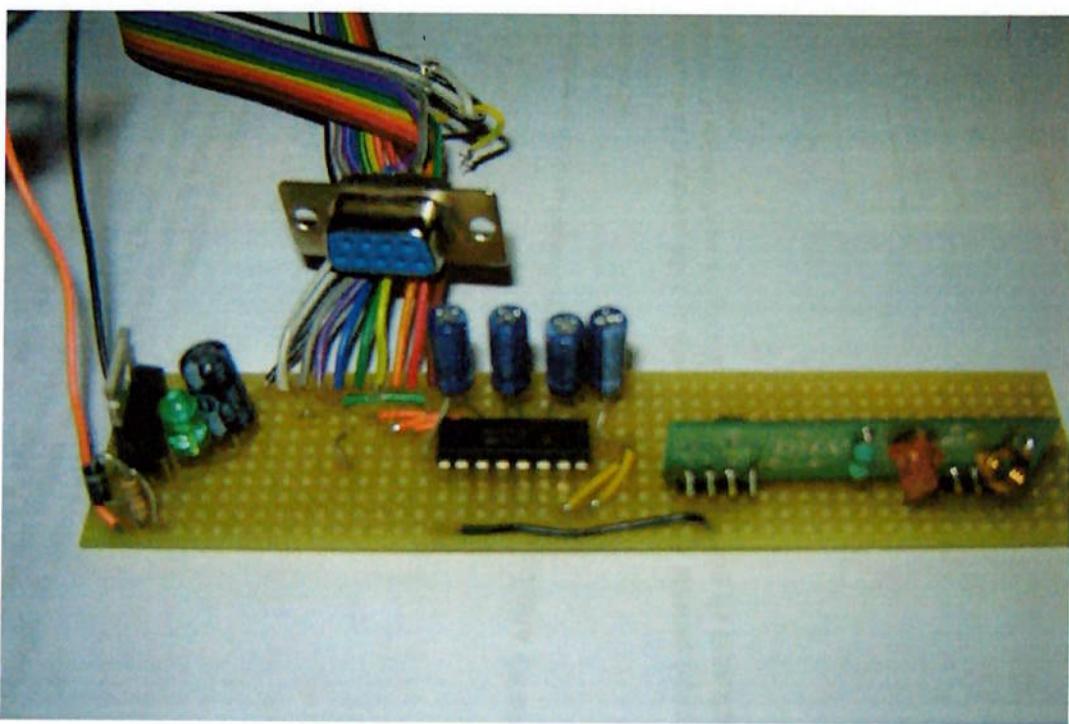


Figura 41 – Foto do receptor

Do ponto de vista técnico do equipamento, obtivemos os resultados que esperávamos. Isso devido ao planejamento que propiciou a formulação e construção dos módulos alem dos testes abundantes dos mesmos. Desta maneira isso confirmou que a subdivisão dos sistemas, no planejamento, foi adequada, visto que os módulos poderiam ser desenvolvidos independentemente e assim testados. Os softwares, por sua vez, apresentaram estruturas robustas e rápidas resultando em nenhuma possibilidade de falha ou travamento foi detectada.

Analizando os resultados obtidos com a comunicação entre o emissor e o receptor foi possível identificar respostas velozes e corretas, mostrando-se a eficiência dos protocolos de comunicação e dos parâmetros de dimensionamento do módulo de radiofreqüência.

Com relação à performance dos dispositivos, depois de realizados alguns testes, foi verificado que este ficou de acordo com os requisitos e parâmetros do projeto; o alcance atingido ficou de acordo com o dimensionado no projeto e a velocidade de interpretação dos sinais está dentro do necessário para a aplicação do equipamento, 1 segundo. Assim, como previsto, o

equipamento possibilita a identificação dos veículos, sem interferência do efeito Doppler e condições adversas do ambiente em um sistema de baixo custo.

8 Conclusão

Ao decorrer do desenvolvimento deste sistema de identificação de linhas de ônibus tentamos elaborar um projeto que visa o atendimento dos requisitos através do gerenciamento dos recursos empregados. Estes recursos, por sua vez foram determinados tanto pelas limitações do sistema quanto pelas condições impostas no projeto do protótipo. Este gerenciamento se deu através do uso dos atributos de engenharia na metodologia que guiou a implantação dos componentes do trabalho. Desta forma, procuramos estabelecer uma relação adequada entre os fatores envolvidos no projeto acadêmico e os requisitos do esquema a ser montado.

Procuramos estruturar nosso trabalho e este relatório em um modelo que permite a adequação do modelo proposto através de avaliações e discussões não só técnicas, mas também sob a perspectiva da sua aplicação. Isso nos permite que a idéia por trás da identificação por Radio Freqüência possa ser também considerada para outras aplicações que não só a indicada neste trabalho.

Nossa aplicação pretende ser eficiente do ponto de vista tecnológico e, desta forma, atender não só o funcionamento de nosso protótipo, mas também futuras adaptações em sua estrutura para possibilitar melhorias em sua utilização. Para tanto o modelo inicialmente proposto para atender as condições necessárias ao devido emprego do sistema foi atualizado ao longo do projeto, tendo em vista a melhoria nos resultados da aplicação deste. Assim, realizamos atividades para o estudo e consequentemente a construção do sistema através de seus conjuntos constituintes.

Buscando a elaboração de um dispositivo tecnológico que seja utilizado em prol da melhoria da qualidade de vida através de uma aplicação social, propiciamos neste trabalho a finalidade devida que a atividade de um engenheiro é capaz de abordar.

Desta forma o projeto como um todo deve ser capaz de apresentar resultados adequados a um trabalho de engenharia, como uma solução para um problema prático, que venha a melhorar as condições de uso do

transporte público nos centros urbanos. Sendo assim, a construção do protótipo corrobora com nossos estudos na constituição deste trabalho final.

9 Referência Bibliográfica

- [Feig] Feig Electronic GmbH, <http://www.feig.de/>, visitado em 30/03/06.
- [Sem Parar] Sem Parar / Via Fácil, <http://www.semparar.com.br/>, visitado em 02/04/06.
- [Auto ID Lab] www.autoidlabs.org, Visitado em 25/04/2006
- [Epc Global] www.epcgobal.com, Visitado em 05/04/2006
- [RFID Journal] www.rfidjournal.com, Visitado em 10/04/2006
- [R-Anatel] RESOLUÇÃO N.º242, Anatel.
- [Anatel] Anatel, <http://www.anatel.gov.br/>, visitado em 02/04/06.
- [Wikipedia] - Wikipedia, <http://www.Wikipedia.org>, visitado em 02/04/06
- [EAN – 2006] – EAN Brasil www.eanbrasil.org.br visitado em 02/04/06
- [POLI] – Poli Cidadã - <http://ubirajara.lac.usp.br/> visitado em 10/04/2006
- [wirelessbrasil]
http://www.wirelessbrasil.org/wirelessbr/secoes/sec_rfid.html Visitado em 05/06/2006
- [ABTU] <http://www.abtu.org.br/> Associação Brasileira de Televisão Universitária, visitado em 10/06/2006
- [RF Aplicações] http://www.computamaps.com/pt_rfplan_applications.php
- [UFG] http://www.eee.ufg.br/cepf/pff/2002/ee_06.pdf - Universidade Federal de Goias, Visitado em 10/06/2006
- [GTA] http://www.gta.ufrj.br/grad/04_2/smartcard/ - Grupo de Teleinformática e Automação, Visitado em 10/06/2006
- [Halliday] Física Volume 1, Halliday – Resnick – Rio de Janeiro, 1976, 1º Edição
- [Souza] David José de Souza, "Desbravando o PIC", 6a Edição, Editora Érica
- [Rogercom] <http://www.rogercom.com/>, RogerCom Pesquisa e Desenvolvimento, Visitado em 20/05/2006
- [R-Fapesp] <http://www.revistapesquisa.fapesp.br> edição 126, Visitado em 10/06/2006

- [Ampères] <http://www.amperesautomation.hpg.ig.com.br/rf.html>, visitado em 05/06/2006
- [rr3XXX] RR3XX Datasheet.
- [rt4XXX] RT4XX Datasheet.
- [RWS] RWS – 371-3 Datasheet.
- [TWS] TWS-AS6 Datasheet.
- [MAXIM] Maxim RS-232 Datasheet.
- [MINICURSO] Minicurso Comunicação serial - RS232
- [ESTV] <http://www.estv.ipv.pt>, Visitado em 01/08/2006
- [ZEGBEE] http://www.rfidgazette.org/2006/07/what_is_zigbees.html, visitado em 30/11/2006
- [BRT]http://www.calccit.org/itsdecision/serv_and_tech/Public_transit_te ch/brt_sum.htm, visitado em 30/11/2006
- [ITRANIT] http://www.iatransit.com/resources/its/wp_2.pdf, visitado em 30/11/2006
- [NATIONAL] <http://www.national.com>, visitado em 30/11/2006

Anexo I – Legislação da ANATEL

De acordo com a legislação devemos atender certas regulamentações visto que: [Anatel] [R-Anatel]

Considerando o disposto no inciso VIII do art. 19 da Lei no 9.472, de 1997, cabe à Anatel administrar o espectro de radiofreqüências, expedindo as respectivas normas;

Considerando deliberação tomada em sua Reunião no 229, realizada em 05 de maio de 2004, resolve:

Art. 1º Republicar, com alterações, o Regulamento sobre Equipamentos de Radiocomunicação de Radiação Restrita, aprovado pela Resolução n.º 305, de 6 de julho de 2002, na forma do Anexo a esta Resolução.

Tendo em vista tais decretos a implantação de enlaces rádio no Brasil necessita de autorização da Anatel. Todavia para os rádios spread spectrum operando em faixas de freqüência de radiação restrita algumas normas específicas são mencionadas.

Assim, de acordo com o “Regulamento Sobre Equipamentos de Rádio Comunicação de Radiação Restrita” emitido pela esta agência temos as definições e restrições que seguem:

Equipamento de Radiocomunicação de Radiação Restrita é definido como: termo genérico aplicado a equipamento, aparelho ou dispositivo, que utilize radiofreqüência para aplicações diversas em que a correspondente emissão produza campo eletromagnético com intensidade dentro dos limites estabelecidos pelo regulamento. Eventualmente, pode estar especificado no Regulamento um valor de potência máxima de transmissão ou de densidade de potência máxima em lugar da intensidade de campo.

Tabela I
 Faixas com restrições de uso

MHz	MHz	MHz	GHz
0,090-0,110	13,36-13,41	399,9-410	5,35-5,46
0,495-0,505	16,42-16,423	608-614	6,65-6,6752
2,1735-2,1905	16,69475-16,69525	952-1215	8,025-8,5
4,125-4,128	16,80425-16,80475	1300-1427	9,0-9,2
4,17725-4,17775	21,87-21,924	1435-1646,5	9,3-9,5
4,20725-4,20775	23,2-23,35	1660-1710	10,6-11,7
6,215-6,218	25,5-25,67	1718,8-1722,2	12,2-12,7
6,26775-6,26825	37,5-38,25	2200-2300	13,25-13,4
6,31175-6,31225	73-74,6	2483,5-2500	14,47-14,5
8,291-8,294	74,8-75,2	2655-2900	15,35-16,2
8,362-8,366	108-138	3260-3267	20,2-21,26
8,37625-8,38675	149,9-150,05	3332-3339	22,01-23,12
8,41425-8,41475	156,52475-156,52525	3345,8-3352,5	23,6-24,0
12,29-12,293	156,7-156,9	4200-4400	31,2-31,8
12,51975-12,52025	242,95-243	4800-5150	36,43-36,5
12,57675-12,57725	322-335,4		Acima de 38,6

Faixa de freqüências (MHz, onde não especificado)	Intensidade de campo (microvolt por metro)	Distância da Medida (metro)
9-490 kHz	2400 F(kHz)	300
490-1705 kHz	24000 F(kHz)	30
1,705-30	30	30
30-88	100	3
88-216	150	3
216-960	200	3
Acima de 960	500	3

Os equipamentos de radiação restrita operando de acordo com o estabelecido no Regulamento devem possuir certificação emitida ou aceita pela Anatel, de acordo com as normas vigentes.

As estações de radiocomunicação, correspondentes a equipamentos de radiação restrita caracterizados por este Regulamento, estão isentas de licenciamento para instalação e funcionamento.

Estas limitações impostas pela agência têm o intuito de estabelecer uma norma que seja atendida por todos os aparelhos em operação e que possa corresponder ao padrão internacional de comunicação por rádio freqüência. Assim, faz-se possível o bom funcionamento dos diversos aparelhos aos quais as normas atendem de forma que se garanta compatibilidade entre os equipamentos e se reduzam interferências causadas pelo uso difundido dos mesmos.