



Centro Universitário de Brasília – UniCEUB
Faculdade de Ciências Exatas e de Tecnologia – FAET
Engenharia da Computação
Giacomo de Almeida Stracquadanio

**TRANSMISSÃO *WIRELESS*: UMA PROPOSTA DE
CONTROLE PARA A FAMÍLIA DE
MICROCONTROLADORES 8051.**

Brasília
2004

Giacomo de Almeida Stracquadanio

**TRANSMISSÃO *WIRELESS*: UMA PROPOSTA DE
CONTROLE PARA A FAMÍLIA DE
MICROCONTROLADORES 8051.**

Projeto de Conclusão do Curso de
Graduação

Engenharia da Computação do Centro
Universitário de Brasília – UniCEUB

Orientadora: Prof^a MC. Maria Marony
Sousa Farias Nascimento

Brasília
2004

Dedicatória

Aos meus pais, Mario e Terezinha, por depositar em mim sua confiança.

À Bianca, pelo amor, paciência e apoio durante todos estes anos.

Agradecimentos

Aos meus pais, Mario e Terezinha, por investirem e insistirem na minha educação.

À Bianca e a os seus pais, Mauro e Suzana, por me apoiar e incentivar nos momentos difíceis.

Aos meus irmãos pela paciência e apoio.

À minha avó Maristela pelo apoio e orgulho durante toda a minha vida e responsável pela minha paixão por computadores.

Aos meus avós Salvatore, Iozzia Giacomina e Guilherme, que são motivo de carinho e admiração.

Ao coordenador do curso de Engenharia de Computação, professor Abiezer Amarília Fernandes, pela seriedade de seu trabalho.

À professora Maria Marony, pela orientação deste projeto.

Ao professor Francisco Javier De Obaldia, por acreditar e incentivar o desenvolvimento deste trabalho.

À todos os professores do curso de Engenharia de Computação que, de algum modo, contribuíram para minha formação durante todos estes anos.

Aos amigos João Thiago, Ricardo Ishida, Marco Aurélio e Daiane, pela amizade e companheirismo.

Enfim, a todos que de uma forma ou de outra contribuíram para a realização deste trabalho.

RESUMO

Neste trabalho, é apresentada uma proposta de um sistema de controle sem fio unidirecional para um microcontrolador da família 8051, utilizando-se módulos de comunicação de rádio frequência, operando em UHF com modulação em FM, como meio de transmissão de dados.

Sistemas que utilizam tecnologia *wireless* trazem muitos benefícios e facilidades que antes não eram possíveis utilizando cabos. A área que trata deste tipo de sistema é fascinante e ainda exige muito estudo e pesquisa. As possibilidades de aplicação são inúmeras.

Por sua vez os microcontroladores da família 8051 são excelentes ferramentas que possibilitam a criação de projetos de automação e controle, incluindo o desenvolvimento de robôs. Embora existam tantos outros controladores com as mesmas funções, o microcontrolador 8051 é um dos mais populares e para o qual observamos o contínuo desenvolvimento de versões com novas funções.

Neste trabalho, são abordadas as teorias de configuração e controle da porta serial do microcomputador, configuração e programação da porta serial do microcontrolador 8051, os programas desenvolvidos em C/C++ e em Assembly, a montagem dos módulos de comunicação e os tipos de antena que podem ser utilizados.

Palavras chaves: **microcontroladores, rádio frequência, wireless.**

SUMÁRIO

INTRODUÇÃO GERAL	1
CAPÍTULO 1: DESCRIÇÃO DO PROJETO DE COMUNICAÇÃO SEM FIO ENTRE UMA ESTAÇÃO DE TRABALHO E UM MICROCONTROLADOR DA FAMÍLIA 8051	4
1.1 INTRODUÇÃO.....	4
1.2 O PROJETO	5
1.2.1 A ESTAÇÃO BASE.....	5
1.2.2 OS MÓDULOS DE RÁDIO FREQUÊNCIA.....	6
1.2.3 A ESTAÇÃO REMOTA	8
CAPÍTULO 2: DESCRIÇÃO E APRESENTAÇÃO DOS PROGRAMAS DESENVOLVIDOS	9
2.1 INTRODUÇÃO.....	9
2.2 O PROTOCOLO	10
2.3 O PROGRAMA DA ESTAÇÃO BASE	11
2.4 O PROGRAMA DA ESTAÇÃO REMOTA.....	15
CAPÍTULO 3: OS MÓDULOS DE TRANSMISSÃO DE DADOS VIA RÁDIO FREQUÊNCIA.....	19
3.1 INTRODUÇÃO.....	19
3.2 OS MÓDULOS DE COMUNICAÇÃO.....	20
3.2.1 O TRANSMISSOR	21
3.2.2 O RECEPTOR.....	23
3.2.3 A ANTENA.....	26
CAPÍTULO 4: RESULTADOS E CONCLUSÕES	29
ANEXO 1: ESQUEMAS DOS CIRCUITOS UTILIZADOS NO PROJETO	31
ANEXO 2: PROGRAMA EM LINGUAGEM ASSEMBLY DA ESTAÇÃO BASE.....	35
BIBLIOGRAFIA	40

Lista de Figuras

Figura 1.1: Modelo do projeto	5
Figura 1.2: Níveis de representação de sinais RS-232.	7
Figura 1.3: Os pinos e circuito interno típico de um MAX-232	7
Figura 2.1: Estrutura do protocolo utilizado na transmissão de dados	10
Figura 2.2: Tela inicial do programa da estação base.	12
Figura 2.3: Tela de configuração da porta serial	13
Figura 2.4: Visão geral do programa da estação remota.	17
Figura 2.5: Tratamento do protocolo pela estação remota.	18
Figura 3.1: Dimensões físicas do transmissor	21
Figura 3.2: Diagrama de blocos do funcionamento do transmissor	22
Figura 3.3: Circuito de teste do transmissor	22
Figura 3.4: Dimensões físicas do receptor	23
Figura 3.5: Diagrama de blocos de funcionamento do receptor	25
Figura 3.6: Circuito padrão do receptor	25
Figura 3.7: Antena helicoidal	27
Figura 3.8: Antena tipo loop	27
Figura 3.9: Antenas tipo chicote	28
Figura A.1: Interface da estação base	32
Figura A.2: Estação remota	33
Figura A.3: Periférico	34

Lista de Símbolos/Definições

API – Applications Programming Interface

ASCII – American Standard Code for Information Interchange

CMOS – Complementary Metal Oxide Semiconductor

DCB – Data Communication Block

EPROM – Erasable Programmable Read-Only Memory

LED – Light Emitting Diode

RS232 – Recommended Standard 232

SAW – Surface Acoustic Wave

SBUF –Serial Data Buffer

SCON – Serial Control

RAM – Random Access Memory

REN – Reception Enable

TTL – Transistor-Transistor Logic

Lista de Tabelas

Tabela 2.1: Valor dos segmentos do protocolo neste projeto.	10
Tabela 3.1: Descrição da função dos pinos do transmissor	21
Tabela 3.2: Descrição da função dos pinos do receptor	24
Tabela 3.3: Avaliação dos tipos de antena	26

Introdução Geral

Para a área de redes de computadores, a utilização de fios e cabos sempre foi uma barreira a ser solucionada. Passar uma quantidade de fios e cabos em locais de difícil acesso sempre torna um projeto mais trabalhoso do que realmente deveria ser. Diversos testes para comprovar a confiabilidade do meio devem ser feitos e, surgindo qualquer problema de comunicação, o primeiro suspeito é a fiação quer seja de cobre ou de fibra ótica. Portanto, algumas soluções foram sugeridas para a substituição desses fios e cabos por diversas empresas e por sociedades científicas, principalmente nos Estados Unidos e na Europa, dando origem ao termo *wireless* ou, literalmente, sem fio.

No entanto, a comunicação sem fio não é uma grande novidade para a maioria das pessoas. Desde a invenção do rádio e da televisão a comunicação *wireless* é bastante difundida e extremamente popular em todas as camadas da sociedade; embora a maioria das pessoas não se dê conta que estão utilizando a tecnologia sem fio. Entre os protagonistas da sedimentação e popularização da televisão temos os satélites (equipamentos de comunicação sem fio, de grande porte e de custo muito elevado) que ajudaram a expandir o alcance tanto da televisão, como também do rádio e da telefonia.

Entre o final do século XX e início do século XXI, o termo *wireless* foi maciçamente empregado para denominar qualquer comunicação sem fio, seja ela utilizando rádio frequência, microondas ou infravermelho, entre aparelhos com poder de processamento, como computadores, telefones celulares e outros. A implantação da comunicação digital sem fio pode trazer para a vida das pessoas acesso a informações sem barreiras. Em qualquer lugar do mundo se pode estar conectado à internet, enviar correios eletrônicos para qualquer parte do mundo, conversar com pessoas em locais muito distantes, ou seja, para a comunicação não existem barreiras.

Neste contexto, este projeto traz a implementação de uma alternativa de comunicação sem fio utilizada para enviar comandos, fazendo uso de programas desenvolvidos especialmente para este fim, a um microcontrolador 8051 que posteriormente acionará um periférico.

Motivação

A tecnologia de comunicação digital sem fio, ou *wireless*, para pequenas distâncias, ou seja, para utilização em escritórios, em indústrias, empresas de diversos ramos do mercado e residências, vêm se tornando mais acessível à medida que seu custo decresce. Um exemplo comum são os inúmeros aeroportos do mundo, incluindo os principais aeroportos do país, que já disponibilizam aos passageiros ou mesmo aos visitantes, acesso à internet, sem que se necessite de um único cabo de rede, bastando, tão somente uma placa própria para redes *wireless* e a assinatura do serviço por alguma provedora de internet local. O serviço traz comodidade e conforto aos que estão viajando a trabalho, oferecendo a oportunidade de estarem conectados enquanto esperam pacientemente o embarque, e alguma diversão àqueles que estão de férias.

Obviamente, esta comodidade não está restrita somente aos aeroportos. Qualquer empresa que adote a tecnologia *wireless* para interligar as suas estações de trabalho, terá esta comodidade. No entanto, a comunicação sem fio digital não fica restrita aos computadores, diversos outros dispositivos podem ou já possuem as características de independência de fios. Poderíamos citar a ampla rede de telefonia celular digital, que é um óbvio e excelente exemplo de modelo.

Diversos tipos de aplicações eletrônicas podem ser beneficiados com a comunicação digital sem fio, inclusive as que utilizam microcontroladores que são excelentes ferramentas em automação para os mais diversos tipos de aplicação. Os microcontroladores atuais já possuem inúmeras utilizações que vão de controle de processos complexos, em indústrias e fábricas, ao controle de um simples acender ou apagar de luzes, em um determinado horário.

Entre os controladores mais utilizados mundialmente, e conhecidos pela sua flexibilidade, está a família dos microcontroladores Intel 8051, voltados basicamente para as aplicações de automação.

Objetivos

O objetivo deste trabalho é estudar e implementar uma comunicação digital de dados, utilizando módulos de transmissão de dados digitais por rádio frequência entre dois sistemas diferentes entre si, representados por um microcomputador e um microcontrolador, que serão chamados respectivamente de estação base, por ser fixa, e de estação remota, pela possibilidade de alteração de posição. Não está sendo estudada a transmissão de dados em

movimento, e sim a característica de portabilidade da estação remota, ou seja, dentro do raio de ação do sistema, a estação remota deve estar parada.

A estação base deve enviar informações que ocasionarão o início de uma ação por parte da estação remota. O dado transmitido será envolto por um protocolo simples de comunicação, que resultará em um pequeno pacote de dados.

Como comprovação do sucesso da transmissão, serão utilizados uma seqüência de leds (diodos emissores de luz) interligados ao microcontrolador, que se acenderão ou irão piscar de maneiras diversas à medida que os comandos enviados pela estação base forem acionados.

Sinopse dos Capítulos

O capítulo 1 trata de um aspecto geral do projeto proposto dando uma visão global dos seus componentes de hardware e uma explicação rápida do funcionamento dos softwares utilizados pelas estações, bem como o seu protocolo. São descritos a estação base e suas características, o que são os módulos de comunicação de rádio frequência e suas principais informações e, por fim, uma breve explanação do que se trata a estação remota. Também é tratado neste capítulo, a necessidade de interfaces entre as estações e como funcionam.

O capítulo 2 traz informações mais detalhadas do funcionamento do protocolo de comunicação, seus segmentos e funções, e o motivo da necessidade desta ferramenta. Também aborda o funcionamento do programa da estação base, explicando o funcionamento do acesso à porta serial no ambiente do sistema operacional utilizado e como será enviado o pacote de dados para a estação remota. Por fim, o capítulo trata do software que irá ser executado pela estação remota, mostrando fluxogramas e incluído informações a respeito do funcionamento da comunicação serial do microcontrolador 8051.

O capítulo 3 descreve a respeito dos módulos de comunicação via rádio frequência, mostra a estrutura interna tanto do transmissor quanto do receptor e as suas características. Informa o modo de como montar a interface com as estações e faz uma análise dos tipos de antena disponíveis e a avaliação de qual tipo será ideal para uso no projeto.

No último capítulo temos a conclusão deste trabalho e, além das considerações finais, novas propostas de desenvolvimento são sugeridas, tendo este trabalho como base.

Capítulo 1

Descrição do Projeto de Comunicação Sem Fio Entre Uma Estação de Trabalho e Um Microcontrolador da Família 8051

1.1 Introdução

A comunicação digital sem fio, por meio de rádio frequência, utilizada neste projeto, se diferencia das tecnologias *wireless* atuais (*Wi-Fi*, *BlueTooth*), por trabalhar em uma faixa de frequência (MHz) bem abaixo das utilizadas atualmente (GHz), o que não prejudica a sua implementação porém, a sua susceptibilidade a interferências é maior. Entretanto, a opção escolhida é mais fácil de ser viabilizada uma vez que seu custo é menor e é mais acessível.

Outro ponto importante são os módulos de rádio frequência utilizados, cujas as características serão apresentadas no capítulo 3, pois não apresentam complexidade em sua implementação. Seu consumo de energia é baixo (entre 12 e 14 mA), ideal para serem utilizados com baterias ou pilhas, o que traz uma característica positiva aos dispositivos que utilizam estes módulos, isto é, a portabilidade. São ideais para aplicações de comunicação digital sem fio, pois todo o tratamento do sinal de comunicação é feito internamente e não há a necessidade da utilização de outros circuitos para tratar a transmissão. O sinal recuperado pelo receptor é tal qual o dado alimentado no transmissor [13].

Para se utilizar do meio de transmissão concedido pelos módulos de comunicação e visando uma futura implementação em sistemas de automação, são usados microcontroladores da família Intel 8051 em conjunto com os receptores, pois são ideais para executar tarefas autônomas. Isto se justifica devido a facilidade de programação, têm um custo baixo e uma grande versatilidade. Também existe muita documentação a respeito destes microcontroladores e uma enorme quantidade de profissionais que se utilizam deles.

A escolha da linguagem de programação Assembly, para utilização nos microcontroladores 8051, se dá pelo fato do código fonte se tornar mais compacto, pois se trata de um código de nível mais baixo em referência a outras linguagens de programação conhecidas. Existem compiladores de linguagem C e BASIC para o microcontrolador 8051, sua programação é mais amigável mas o código fonte final se torna muito mais extenso, ocupando um espaço na memória que poderia ser melhor utilizado.

1.2 O Projeto

Este estudo propõe que seja realizada uma transferência de dados digitais, utilizando rádio frequência, entre um computador pessoal, que agora denominamos de estação base, e um microcontrolador da família Intel 8051 (denominado estação remota) que controlará um periférico qualquer, ligado diretamente a ele. A figura 1.1 mostra um modelo do projeto.

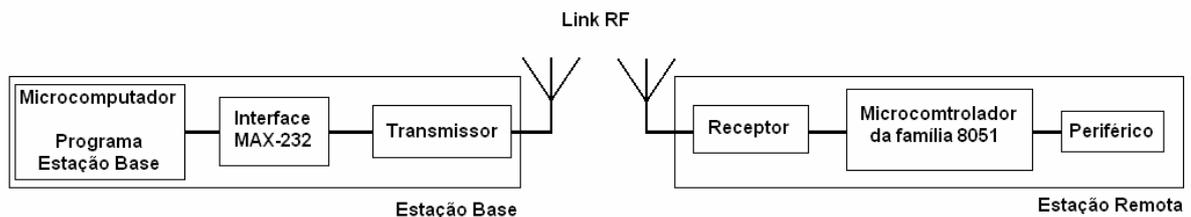


Figura 1.1: Modelo do projeto

O tipo de transmissão dos dados é projetado para ser simplex, ou seja, a transmissão é feita somente em uma única direção. É possível realizar transferências de dados do tipo full-duplex, envio e recebimento de dados ao mesmo tempo. Para isso, se deve substituir os módulos receptor e transmissor por um par de transceptores, que são módulos de comunicação bidirecional, mas tem um custo mais elevado.

1.2.1 A Estação Base

A estação base, como descrita anteriormente, é um computador de uso comum sem um grande poder de processamento, mas que esteja apto a rodar o sistema operacional *Windows* da empresa *Microsoft*, em sua versão de 32 bits (Win32). O sistema operacional mais antigo que utiliza o padrão Win32 é o *Windows 95* e o *Windows NT*. Os requisitos mínimos e necessários para instalar os sistemas operacionais são:

- Processador 386DX ou versão mais recente (486 recomendado)
- 4 MB de memória RAM (8MB recomendados)
- Espaço em disco rígido de 50 a 55 MB
- Um drive de disquete de 3,5 polegadas de alta densidade
- Monitor VGA ou com melhor resolução (SVGA colorido recomendado)

Como será utilizada a linguagem C/C++, para a criação do software, devemos levar em conta a seguinte restrição deste sistema operacional: o acesso direto ao hardware do

computador geralmente não é permitido. As alternativas para desenvolver aplicações que se utilizam da comunicação da porta serial estão no uso das funções de comunicação da *API* (*Applications Programming Interface*) do sistema operacional. Os antigos comandos para comunicação através de hardware utilizados pela linguagem C/C++, tais como *inport*, *inpw* e *outport*, não são mais suportados e estão fora de uso [1]. Portanto, para criar o programa, é necessário conhecer como o sistema operacional concede permissão aos recursos de *hardware*, o que será apresentado mais adiante.

Para facilitar a interação do usuário com o programa da estação base e para facilitar o acesso aos recursos do sistema operacional, será utilizado o compilador C++ *Builder* versão 6, da empresa *Borland Software Corporation*, pois facilita a criação de uma interface gráfica familiar e padronizada com o ambiente utilizado.

O programa da estação base não foi projetado para enviar os dados somente, os comandos que serão transmitidos à estação remota, serão envoltos em um protocolo de transferência. O objetivo e função deste protocolo é evitar que, na presença de mais de uma estação remota, as informações sejam enviadas a destinos equivocados. O protocolo é formado por campos que contêm:

- campo que indica início do pacote;
- campo de endereço do destino;
- campo de endereço da origem;
- campo de dados;

Portanto, antes da transmissão dos dados para o microcontrolador, estes serão devidamente encapsulados pelo protocolo e em seguida enviados para a porta serial. Não existe a presença de criptografia ou de algum mecanismo que possa identificar um erro na transmissão, então se algum bit for modificado durante a transmissão, o microcontrolador não irá reconhecer o dado e descartará o pacote.

1.2.2 Os Módulos de Rádio Frequência

Os módulos de rádio frequência devem ser conectados à porta serial do computador ou da estação base. Esta interligação não pode ser feita diretamente, deve haver uma interface de conversão, pois todos os componentes envolvidos utilizam níveis lógicos de tensão TTL/CMOS, ou seja, trabalham em níveis de tensão de 0V (zero volts) para representar o bit de valor '0' (zero) e 5V (cinco volts) para representar o bit de valor '1' (um). Já a porta serial,

utiliza para representar o bit de valor '0', o intervalo de tensão entre +3V à +25V, e para representar o bit de valor '1', o intervalo de tensão entre -3V à -25V [12], como apresentado na figura 1.2.

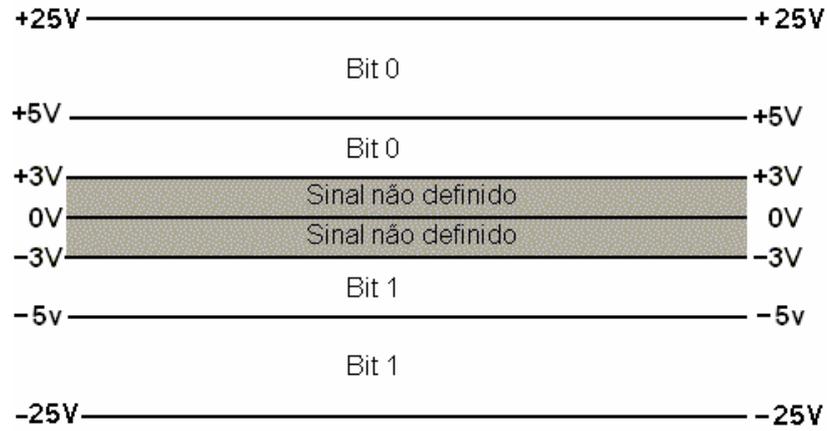


Figura 1.2: Níveis de representação de sinais RS-232.

Portanto, temos uma incompatibilidade entre as tensões da porta serial e as tensões utilizadas pelos módulos de rádio frequência e pela estação remota.

Como solução para esta incompatibilidade será utilizado um circuito integrado, chamado comercialmente de MAX-232, responsável pelas conversões necessárias para a comunicação entre componentes TTL/CMOS com a porta serial. A figura 1.3 mostra um exemplo de um circuito interno típico do MAX-232.

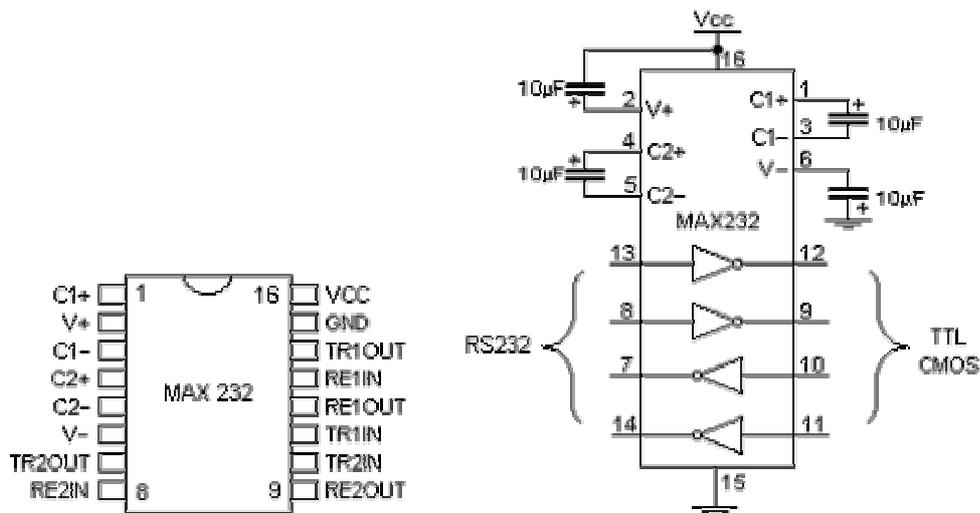


Figura 1.3: Os pinos e circuito interno típico de um MAX-232 [12]

Tanto o módulo de rádio frequência transmissor quanto o módulo de rádio frequência receptor utilizados, são placas de circuito impresso de tamanho miniaturizado. Eles

transmitem em UHF (*Ultra High Frequency*) a uma frequência de 418MHz, e utilizam FM (*Frequency Modulation*) para a modulação do sinal. São ideais para implementações simples de transmissão de dados e possuem uma taxa de transferência de, aproximadamente, 40kbit/s. Têm alcance de comunicação a uma distância de aproximadamente 75m em locais fechados e por volta de 300m em ambientes abertos. Todo o sinal que o transmissor recebe da estação base é enviado via rádio frequência. Este é acolhido pelo receptor que transforma as ondas em sinais elétricos e as envia à estação remota [13]. Não foram feitos estudos a respeito do grau da ação de interferências no projeto, por não este o foco do estudo.

Ao contrário do módulo transmissor, neste caso, o módulo receptor não necessita de nenhuma interface para se conectar à estação remota, pois ambos possuem as mesmas características de representação de bits: tensão de 0V é representado pelo bit '0' e tensão de 5V representado pelo bit '1', ou seja, trabalham com tensões de circuitos TTL/CMOS. Portanto, dispensa o uso da interface MAX-232 da estação remota.

1.2.3 A Estação Remota

A estação remota utilizada neste projeto é o protótipo de uma placa de testes para programação, de uso genérico, do microcontrolador da família 8051. Será utilizada a linguagem *Assembly* para criar o programa que estará alocado na memória do microcontrolador.

O microcontrolador tem a característica de controlar um periférico, ou um processo, portanto, ao receber a informação da estação base, o microcontrolador irá executar uma ordem de comando que acionará um periférico ligado a ele. Este periférico pode ser tanto um motor de passo, uma seqüência de LEDs, um *display* de cristal líquido, e muitos outros. Ou ainda, podem existir sensores ligados ao microcontrolador tais como sensores de luminosidade, de temperatura, de pressão, de ondas sonoras, etc., que irão fornecer dados para que o microcontrolador execute alguma tarefa pré-estabelecida. Assim, o microcontrolador recebe as informações e executa as tarefas conforme foi programado.

Como o microcontrolador não receberá somente dados simples mas encapsulados por um protocolo, é tarefa também do microcontrolador receber estes dados, envoltos no protocolo, e separá-los, julgando se é ou não início de transmissão de dados; verificar se é o endereço ao qual é destinado aquele dado e se está autorizado a receber informações da estação base que transmitiu os dados. Só depois de verificar todas estas questões, irá armazenar o dado transferido e encerrar a recepção dos dados para, em seguida, executar o comando recebido.

Capítulo 2

Descrição e Apresentação dos Programas Desenvolvidos

2.1 Introdução

Neste capítulo, iremos abordar um dos importantes pontos do projeto: os programas. Será descrito como foi montada a sua estrutura, como se dá o seu funcionamento e suas funções. Existem dois programas para este projeto. Um escrito em *C/C++*, desenvolvido para trabalhar com a estação base, e um outro programa escrito em *Assembly*, para trabalhar na estação remota.

A montagem dos códigos dos dois programas é totalmente voltada para a programação e configuração da porta serial. Mas, quando tratamos da estação base, uma peculiaridade do sistema operacional utilizado não permite o acesso direto ao *hardware* de comunicação disponível, portanto, outras soluções foram adotadas tais como: a utilização das APIs do sistema operacional e utilização de *drivers* escritos exclusivamente para a porta serial, de código fonte aberto [1]. Já a estação remota não apresenta maiores dificuldades de acesso e pode ser programada e configurada sem dificuldades.

Outro tópico importante é o protocolo utilizado. Não foi adotado nenhum protocolo já existente pois, como se trata de uma estrutura pouco complexa, criar um próprio torna o projeto mais maleável e permite maiores contribuições futuras. Apesar da simplicidade, o protocolo utilizado na transferência de dados é uma ferramenta fundamental para assegurar a individualidade da estação remota, isto é, em presença de mais de uma estação remota, somente uma irá responder aos comandos enviados. Caso seja necessário que as outras estações remotas executem o mesmo comando, a estação base terá que enviar o mesmo número de comandos quantos forem o número de estações remotas.

2.2 O Protocolo

Como dito anteriormente, para melhorar a condição de controle da estação remota, um protocolo simples, mas funcional, será adotado. Trata-se de um pacote formado de quatro segmentos, como mostrado na figura 2.1.



Figura 2.1: Estrutura do protocolo utilizado na transmissão de dados

Cada segmento é composto por uma palavra de um byte, formando então um pacote de quatro bytes. Cada segmento é um valor pré-definido igual tanto para a estação base quanto para a estação remota. respeitando as indicações da tabela a seguir:

Tabela 2.1: Valor dos segmentos do protocolo neste projeto.

Função do segmento	Valor em hexadecimal	Valor em ASCII
Início do pacote	4C	L
Endereço de destino	77	
Endereço de origem	51	Q
Dados		
Acender pares	2A	*
Acender ímpares	3B	;
Acender todos	4C	L
Piscar pares	42	B
Piscar ímpares	6E	N
Piscar todos	50	P

Início do pacote

O primeiro segmento é aquele que indica o início do pacote. A estação base deve incluir este segmento antes de enviar o pacote e a estação remota deve reconhecer a sua função quando recebê-lo.

O segundo segmento do pacote leva a informação a respeito da estação remota a que se destina. Outras estações remotas podem, eventualmente, receber a informação de início de pacote, mas não devem aceitar o resto dos dados se não forem configurados com o endereço do destino correto.

O próximo segmento contém informações a respeito do endereço de origem do pacote. Este segmento é utilizado pelas estações remotas para verificar se podem receber informações da estação base, que está enviando os pacotes de dados. O objetivo deste segmento é evitar

que haja qualquer estação base, estranha ao ambiente, que tenha comando imediato às estações remotas.

Por último, e o mais importante, vem o segmento contendo o dado com as informações a respeito da ordem enviada pela estação base à estação remota. Como se pode notar, não existe um segmento que tenha a função de “fim de pacote”. Cada parte do protocolo tem um tamanho fixo (8 bits), a estação remota, além de conter os dados, também finaliza o protocolo, pois reconhece no último segmento o fim da transmissão, ou seja, o programa que irá tratar o protocolo irá aguardar quatro segmentos de um byte e então finaliza o processo.

2.3 O Programa da Estação Base

O programa da estação base é desenvolvido em linguagem C/C++ e tem uma interface gráfica simples e intuitiva. A sua função principal é colher a ordem do usuário, montar o protocolo e enviar, via porta serial do computador, o pacote. A comunicação serial é feita utilizando a API do sistema operacional, e necessita de conhecimentos de programação avançada.

Para o usuário, existem as seguintes opções de configuração:

- Configurar pacote.
 - Início do pacote (qual palavra será utilizada para indicar início de pacote).
 - Dados (numerados de um a seis que representam cada uma das ordens de comando disponíveis).
- Configurar endereços.
 - Endereço da estação base.
 - Tabela de endereços de estações remotas.
- Configurar porta serial.
 - Porta de comunicação.
 - Taxa de transferência.
 - Número de bits de dados.
 - Número de bits de parada.
 - Paridade.
 - Controle de fluxo.

Para as ordens de comando, existem as seguintes opções:

- Conectar/Desconectar porta serial
- Acender leds em formação par.

- Acender leds em formação ímpar.
- Acender todos os leds.
- Piscar leds em formação par.
- Piscar leds em formação ímpar.
- Piscar todos os leds.

Cada uma destas opções possui um botão independente configurado no programa. Assim que forem acionados pelo *mouse* do computador, o pacote é montado e enviado automaticamente.

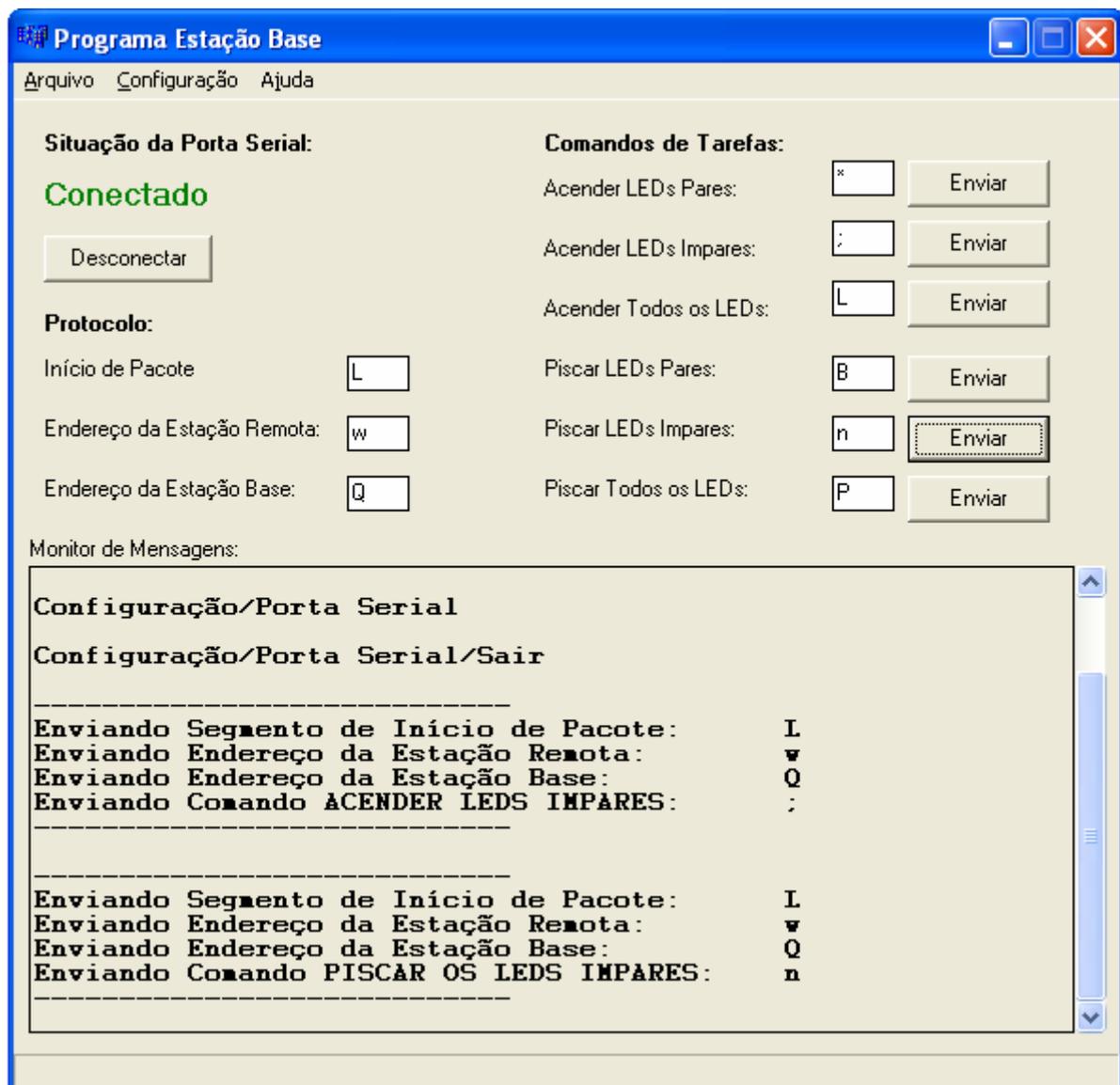


Figura 2.2: Tela inicial do programa da estação base.

Para montar o protocolo de transmissão, o programa simplesmente irá enviar à porta serial um byte por vez respeitando um intervalo de tempo e no formato de valores hexadecimais. Portanto, será enviado primeiro o byte de início de pacote, depois o byte de endereço estação remota, em seguida o byte de endereço da estação base e por fim o byte de dado, ou seja, o comando de tarefa.

O intervalo de tempo proposto é necessário para que o protocolo seja enviado sem que a estação remota perca alguma parte do pacote por falta de tempo de processamento. O intervalo de tempo é muito curto, não ultrapassando 1,0 segundo.

As configurações da porta serial podem ser alteradas, mas não devem fugir à configuração padrão para evitar problemas de comunicação durante a transmissão de dados. A configuração do protocolo está totalmente aberta e pode ser alterada a qualquer momento (figura 2.3), pois o objetivo do programa neste primeiro momento é de se fazer testes e demonstrações.

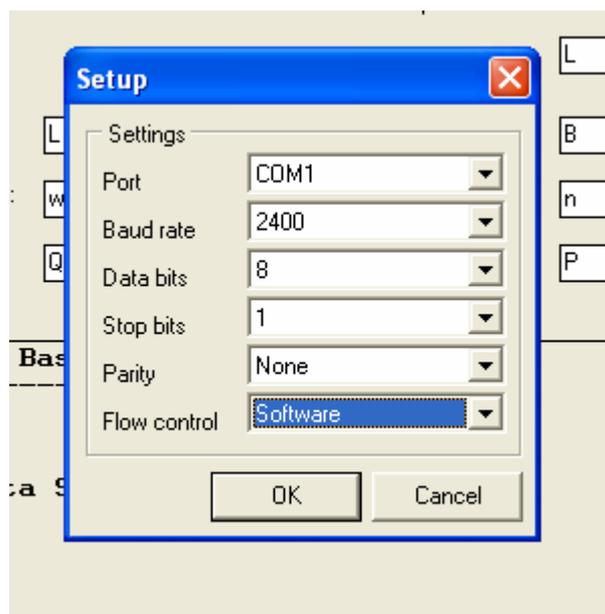


Figura 2.3: Tela de configuração da porta serial.

Para que o programa funcione de acordo com o esperado, todos os programas que utilizam a porta serial devem ser encerrados.

Para ter acesso às funções da porta serial no sistema operacional utilizando a API, é necessário que seja iniciada a porta serial utilizando o comando *CreateFile* abrindo um *Handle* [3], como no exemplo a seguir:

```

HANDLE hComm = NULL;

hComm = CreateFile("COM1",           //porta a ser utilizada
                  GENERIC_READ | GENERIC_WRITE,
                  0,                  //acesso exclusivo
                  0,                  //ausência de segurança
                  OPEN_EXISTING,
                  0,                  //sem I/O overlapped
                  0);                 //sem template

```

Portanto, ao adquirir acesso à porta serial, é escolhida a porta COM1 do computador para uso exclusivo do programa, a configuramos para que tenha permissão para escrita e leitura de dados, sem protocolo de segurança e não utilize estrutura de entrada e saída (*overlapped* - a comunicação não precisa aguardar resposta para completar uma operação); a função de acesso à porta serial não é um modelo genérico [3].

As configurações são carregadas na iniciação do programa, sendo definidos os seguintes parâmetros:

```

bPortReady = GetCommState(m_hCom, &m_dcb);
dcb.BaudRate = 9600;
dcb.ByteSize = 8;
dcb.Parity = NOPARITY;
dcb.StopBits = ONESTOPBIT;
dcb.fAbortOnError = TRUE;

```

As configurações da porta serial são especificadas por um *Data Communication Block* (DCB) ou, literalmente, bloco de dados de comunicação. A forma mais comum de iniciar um DCB, é fazendo uma chamada à função *GetCommState* e incluindo os parâmetros. Os valores configurados são valores padrão para o funcionamento, o mais simples possível, da porta serial. Portanto, temos a taxa de transmissão a 9600 bits por segundo, o tamanho do byte transmitido de 8 bits, ausência de paridade, um bit de parada e, em caso de erro, a transmissão será cancelada [3].

Incluir e configurar o tempo de resposta (*timeout*) da porta serial é opcional mas pode ser configurado da mesma maneira que o DCB, porém, não será utilizado neste programa pelo simples fato do programa base não esperar qualquer resposta vinda da porta serial, o programa só envia mensagens [3].

Finalizada a configuração da porta serial, já é permitido iniciar a função para enviar um dado para a porta serial em formato hexadecimal, utilizando a seguinte linha de código:

```

TransmitCommChar(hComm, <palavra>);

```

No caso deste programa, cada botão é responsável pelo envio um comando diferente. Está preparado para ler qual o caractere em ASCII está escrito nas caixas de texto e transmite a informação em formato hexadecimal assim que o botão for acionado. A conversão de ASCII para hexadecimal é muito importante para que a estação remota possa interpretar corretamente a ordem enviada.

2.4 O Programa da Estação Remota

O programa da estação remota pode ser dividido em três fases. A primeira é a configuração da interface serial do microcontrolador, a segunda é o desmanche do protocolo enviado pela estação base e a terceira, e última fase, é a execução da ordem enviada.

Na primeira fase, o microcontrolador carrega, em seus registradores, os valores que serão necessários para que se possa ter acesso à porta serial. São eles o registrador **SBUF** e o registrador **SCON**. O registrador **SBUF** é responsável pela entrada e saída de dados do microcontrolador 8051, utilizando comunicação serial. É formado por dois bytes, um para transmissão e outro para a recepção dos dados, sempre utilizando o mesmo endereço de acesso. Já o registrador **SCON**, é responsável pelo controle da interface serial e tem o tamanho de um byte. Cada bit do **SCON** tem uma função de controle e são divididos da seguinte maneira: os três primeiros bits são denominados **SM0**, **SM1** e **SM2** e as combinações entre eles determinam o tipo e modo de operação da porta serial. É gerada uma interrupção e pode ser habilitada a comunicação entre outros microcontroladores 8051. O quarto bit, denominado de **REN**, é responsável pela habilitação da recepção de dados da porta serial. Os bits seguintes têm, cada um, uma função isolada mas não terão utilidade nesta aplicação, por isso, estarão desativados [10, 11].

A seleção do modo de operação determinada pelos bits **SM0** e **SM1** podem ser de quatro modos diferentes. O modo que será utilizado, e que atende melhor ao tipo da aplicação, é o modo 1. Neste modo é configurado o bit **SM0** como 0, e o bit **SM1** como 1, que determinam que o modo de transmissão seja do tipo assíncrono e que a taxa de transmissão pode ser variada e controlada pelo bit **SMOD** da palavra **PCON** e pelo **T/C1**, obedecendo às equações a seguir:

$$\text{Taxa de transmissão} = (\text{taxa de overflow do T/C1})/32, \text{ para SMOD} = 0$$

$$\text{Taxa de transmissão} = (\text{taxa de overflow do T/C1})/16, \text{ para SMOD} = 1$$

Entretanto, como se trata de uma aplicação somente para recepção de dados, esta decisão somente será de alguma utilidade para uma futura aplicação que possa se utilizar das

características de transmissão de dados do microcontrolador. Se não houvesse este interesse, qualquer um dos modos de transmissão poderia ser utilizado [10, 11].

Depois de configurada a porta serial do microcontrolador, uma tabela de endereços é carregada. A tabela é formada, primeiro pelo endereço do microcontrolador, em seguida pelos endereços das estações bases que terão acesso à estação remota. Cada um deles, tanto da estação base e da estação remota, possui um endereço do tamanho de um byte, ou seja, oito bits.

Em seguida, uma função de espera de informações é carregada (figura 2.2). Esta função estará aguardando que alguma informação seja recebida pela porta serial. Assim que a recebe, uma outra função reservada para o tratamento do protocolo utilizado é acionada e o primeiro byte que é recebido é comparado com o valor de início de pacote. Se for igual, segue para o próximo passo. Se não for igual, o byte é descartado e é aguardada a próxima informação. O próximo passo, depois de validado o início de pacote, é aguardar o próximo byte.

Assim que é recebida, ainda na função de tratamento de protocolo, a informação é comparada com o endereço da estação remota. Se o endereço for válido, então a informação pertence à estação remota; se não for aceita, a informação é descartada e os próximos bytes serão ignorados.

Depois de verificar se o protocolo pertence à estação remota, o programa verifica se a informação é de uma origem permitida. Portanto, depois de verificar se o endereço da estação remota é válido, o programa aguarda o próximo byte e consulta a tabela de endereços e verifica se o endereço da estação base é válido. Se for verdadeiro, o programa segue para a próxima etapa. Se não for um endereço válido o próximo byte é ignorado.

Como já foram feitas as análises de endereçamento, esta informação pertence à estação remota e foi enviada por uma estação válida. O programa da estação remota então espera o próximo byte a ser recebido, armazena a informação e finaliza o protocolo, pois o tamanho esperado pelo programa para executar o comando é de um byte apenas.

O dado recebido é armazenado e o programa base analisa e executa a tarefa ordenada pela estação base. O programa irá executar a ordem até que novas informações válidas sejam recebidas.

Podemos ver a seguir no diagrama de blocos demonstrado na figura 2.4, em que um deles é a visão geral do funcionamento do programa, e o seguinte diagrama de blocos demonstrado na figura 2.5 apresenta uma visão mais aprofundada e estendida da função de tratamento de protocolo.

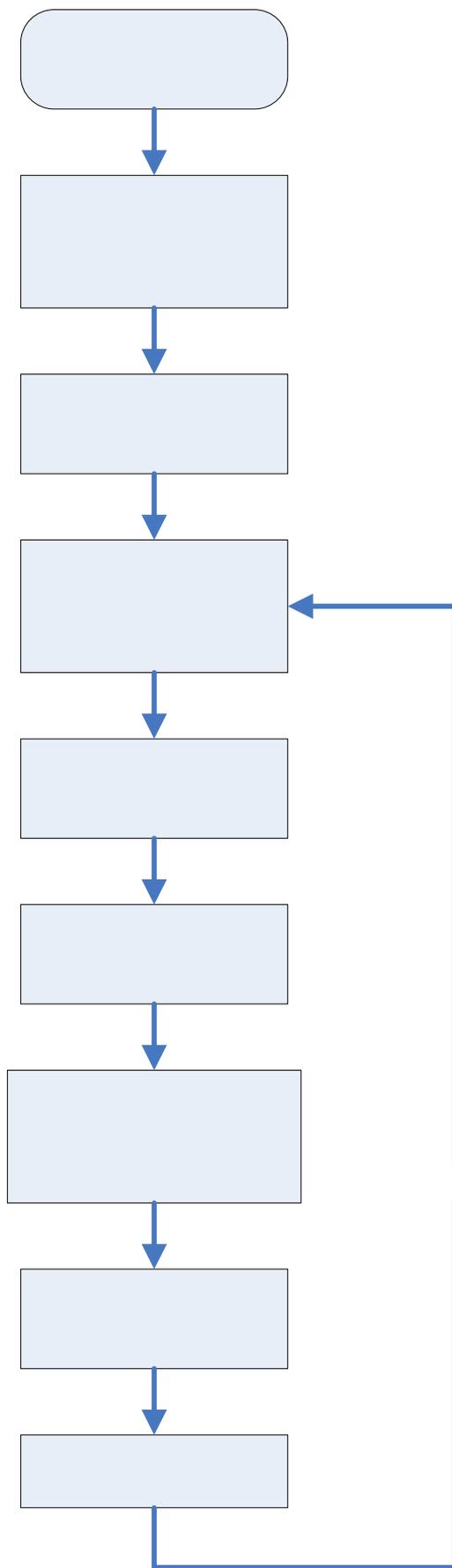


Figura 2.4: Visão geral do programa da estação remota.

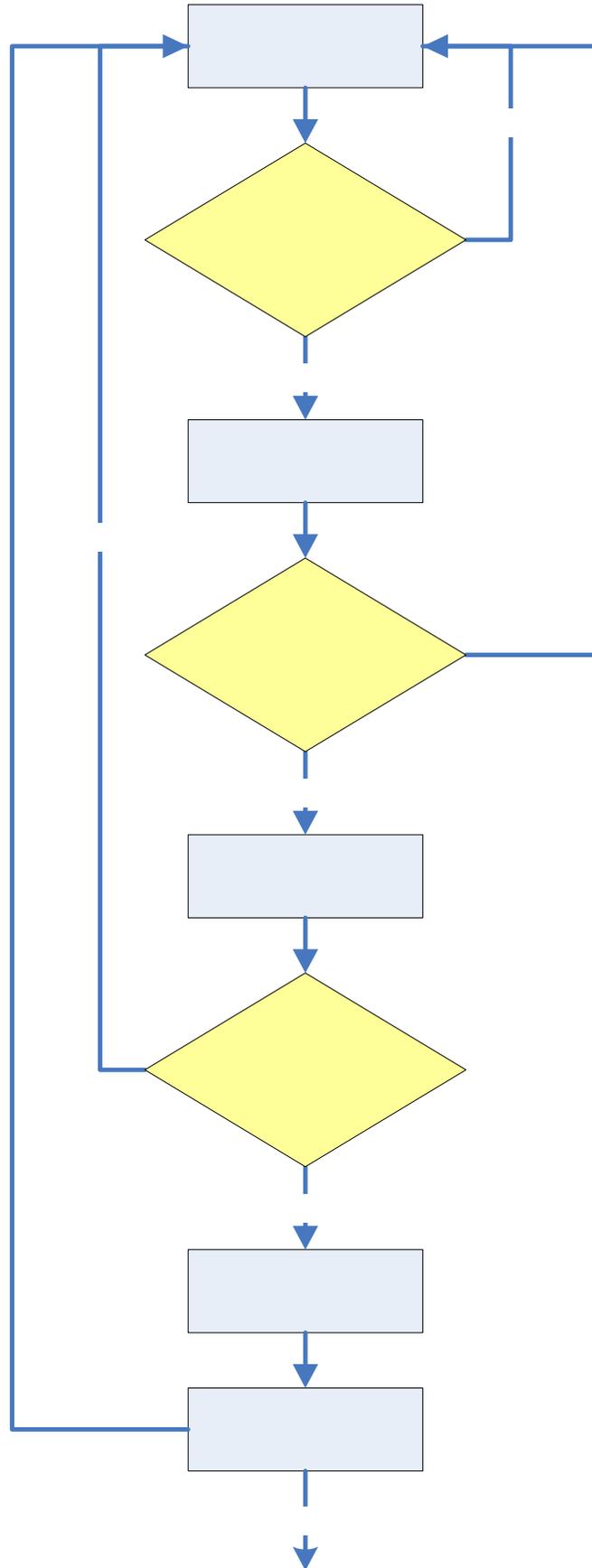


Figura 2.5: Tratamento do protocolo pela estação remota.

Capítulo 3

Os Módulos de Transmissão de Dados Via Rádio Freqüência.

3.1 Introdução

Neste capítulo será abordada uma parte mais importantes e também a mais delicada deste projeto. O meio de comunicação escolhido, a utilização de ondas eletromagnéticas emitidas pelo espaço para transmitir informações digitais, não é livre de interferências externas e está sujeito a desvios de alterações nas ondas emitidas pelo transmissor que podem ser gerados por eletrodomésticos, alguns tipos de lâmpadas, enfim, por outro equipamento funcionando na mesma freqüência e que não faça parte do meio [6,7].

Neste projeto é utilizado um par de módulos de comunicação, um transmissor (TX) e um receptor (RX), que estabeleceram um contato (*link*) através de ondas de rádio, concretizando a transmissão dos dados da estação base para a estação remota. Estes módulos foram montados e desenvolvidos pela empresa inglesa *Radiometrix*, portanto todos os valores de resistências, capacitores e outros componentes, foram orientações diretas do manual de utilização do equipamento [13].

Os módulos de comunicação foram desenvolvidos especialmente para o tipo de aplicação a que este projeto se destina: controle de um sistema à distância com a ausência de fios ou cabos de comunicação (*wireless*). Podem ser utilizados para aplicações independentes, isoladas, ou em aplicações voltadas para a criação de redes. Têm tamanhos reduzidos e podem ser usados em aplicações portáteis, pois o consumo de energia é muito baixo [13].

Uma parte importante na implementação dos módulos de comunicação é a escolha do tipo de antena que atenda à necessidade da aplicação. Nessa escolha, deve ser levado em consideração o ambiente em que será utilizado o equipamento, o nível de ruído do ambiente, o alcance necessário para o sucesso da comunicação e necessário para a aplicação. Outros detalhes serão apresentados mais adiante neste capítulo.

3.2 Os Módulos de Comunicação

Os módulos de comunicação nada mais são que pequenas placas de circuito impresso. Nelas existem circuitos moduladores (no transmissor) e demoduladores (no receptor) de FM que trabalham na faixa de frequência UHF (*Ultra High Frequency*) [13] também conhecida como frequências ultra-altas ou ainda ondas muito curtas. Segundo o Quadro de Atribuições de Frequências no Brasil fornecido pela Agência Nacional de Telecomunicações (ANATEL), a distribuição dentro do espectro de frequências para as ondas UHF vai de 300MHz a 3,0GHz sendo que a utilizada neste projeto é de 418MHz.

Tanto o transmissor quanto o receptor utilizam tecnologia de ondas tipo SAW (*Surface Acoustic Wave* ou Onda Acústica Superficial) para o tratamento interno do sinal. As razões para a utilização desta tecnologia são: o fornecimento de altas frequências com exatidão e a excelente imunidade a alterações de frequência em razão da temperatura ou da idade do equipamento. Outra vantagem é a relativa imunidade a ruídos e a excelente sensibilidade à recepção de sinais [6, 7].

3.2.1 O Transmissor

É o componente do projeto responsável pela transmissão de dados, utilizando ondas de rádio operando na faixa de frequência 418MHz, para o meio externo ao sistema. O transmissor utiliza modulação por frequência (FM) para converter os sinais digitais do computador em sinais de rádio, utilizando tecnologia SAW. Sua tensão de operação está entre 2,2V e 6,0V, dependendo da versão do transmissor. Neste projeto o módulo TX opera entre 4,0V e 6,0V e se seu consumo é de 12mA [13].

Na figura 3.1, é apresentada a dimensão física do transmissor e a numeração de seus pinos. Dos cinco pinos existentes os mais importantes, do ponto de vista da transmissão de dados, são os pinos 2 e 5, respectivamente a saída para a antena e a porta de entrada de dados. Os outros pinos estão ligados à alimentação do componente [13].

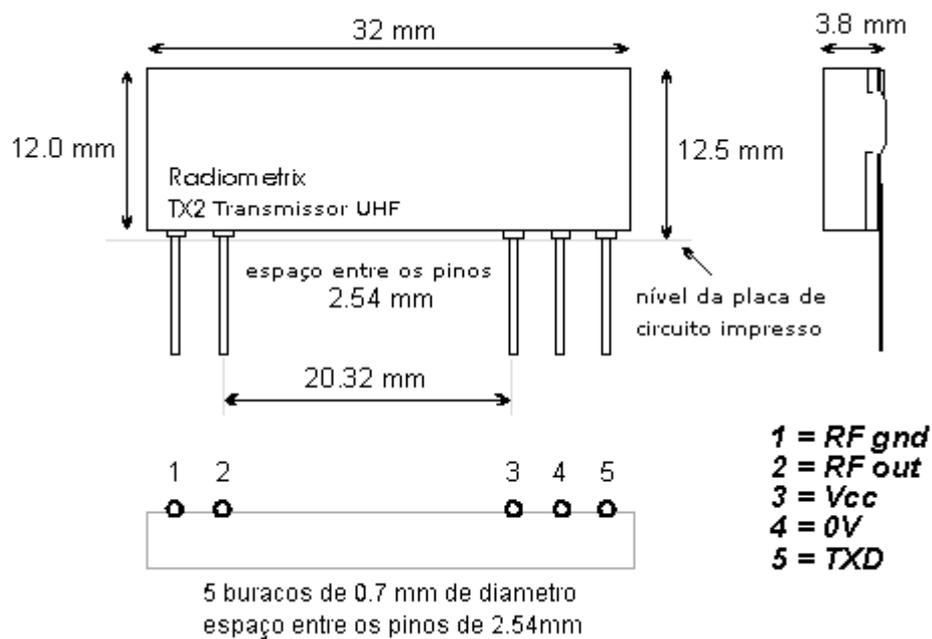


Figura 3.1: Dimensões físicas do transmissor [13].

A tabela a seguir, descreve a função de cada um dos pinos do transmissor.

Tabela 3.1: Descrição da função dos pinos do transmissor [13]

Número do Pino	Nome	Descrição
1	RF GND	Terra RF
2	RF OUT	Saída RF de 50Ω para a antena. Isolado internamente.
3	Vcc	Pino de alimentação do transmissor. O módulo irá gerar RF quando a alimentação estiver presente. Sugere-se a adição de um capacitor cerâmico de desacoplamento de 100nF.
4	0V	Conexão com o terra do sistema.
5	TXD	Entrada de dados, aceita dados seriais digitais, respeitando os níveis de tensão de 0V à Vcc ou sinais lineares de alto nível. Impedância de entrada de 100kΩ.

A figura 3.2 mostra um esquema de como funciona a modulação dos dados. O sinal digital que entra pelo TXD, pino 5, passa por um filtro passa-baixa de 100kHz de segunda ordem, para filtrar o sinal vindo do exterior. Em seguida, é acoplado a um oscilador estabilizado tipo SAW, passa por um *buffer* e, em seguida, o sinal é tratado por um filtro

passa-banda de 418MHz. Depois o sinal é enviado pela saída RF OUT, pino 2, que está ligado a uma antena [13].

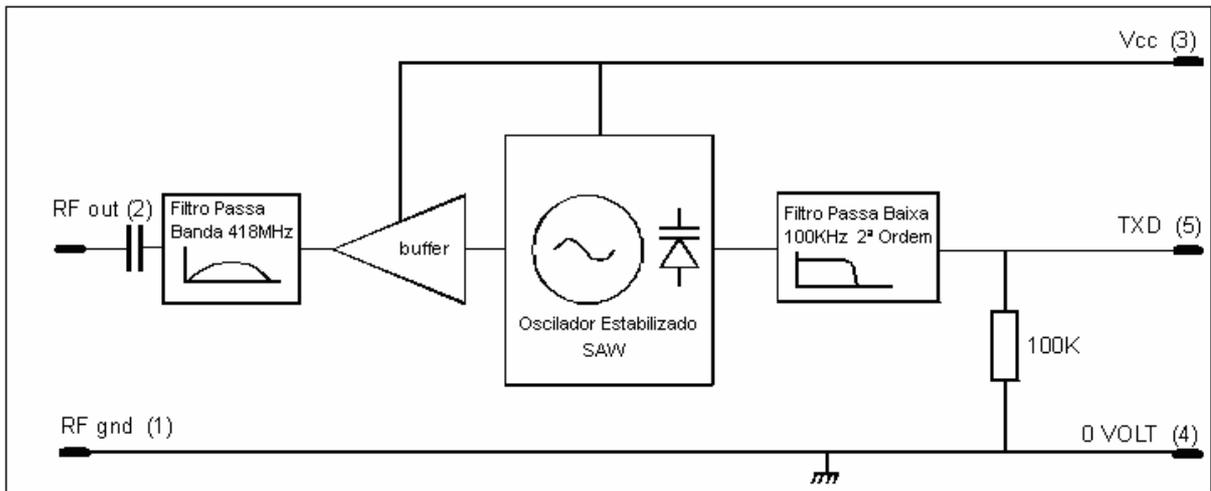


Figura 3.2: Diagrama de blocos do funcionamento do transmissor [13]

A montagem básica para a operação do transmissor é apresentada na figura 3.3. Os pinos 1 e 4 são ligados diretamente ao terra do sistema (*GND*). O pino 2 se conecta à antena do transmissor de valor de 50Ω . O pino 3, que recebe a alimentação do sistema, deve ter um capacitor de acoplamento de 100nF . Já o pino 5 pode ser ligado diretamente à fonte de emissão de dados (estação base) [13].

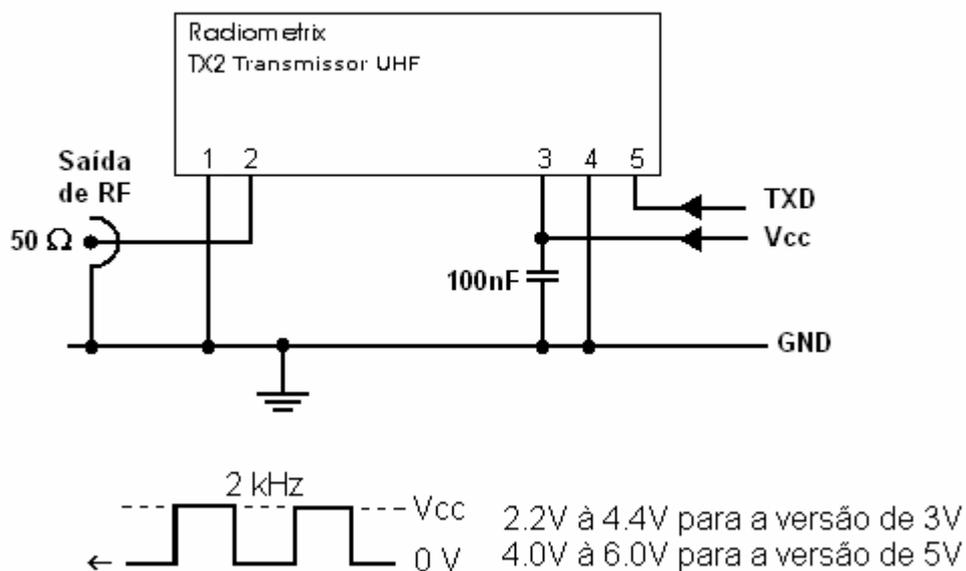


Figura 3.3: Circuito de teste do transmissor [13]

O transmissor pode ser facilmente implementado, tendo o cuidado com o tipo de tensão de entrada dos dados. Como foi dito no primeiro capítulo, é necessária a presença de uma interface RS-232 que faça a mudança de tensões de nível de porta serial para tensões de nível CMOS/TTL.

3.2.2 O Receptor

O receptor é o componente que estará ligado à estação remota do sistema, trabalha na mesma frequência do transmissor e capta as informações provenientes do meio externo enviadas por ondas de rádio na faixa de frequência de UHF. Opera em tensões entre 3V a 6,0V, dependendo da versão, e seu consumo é de 14mA. É capaz de receber informações a uma taxa de 40kbps [13].

Embora tenha dimensões maiores em comparação ao transmissor, sua portabilidade não é comprometida e ainda é ideal para sistemas portáteis. Existem 7 pinos dos quais, 3 (os pinos 2, 4 e 5) são responsáveis pela alimentação do receptor. Os outros 4 pinos são responsáveis pela saída de dados recebidos. A figura 3.4 mostra as medidas das dimensões do receptor e a numeração de seus pinos.

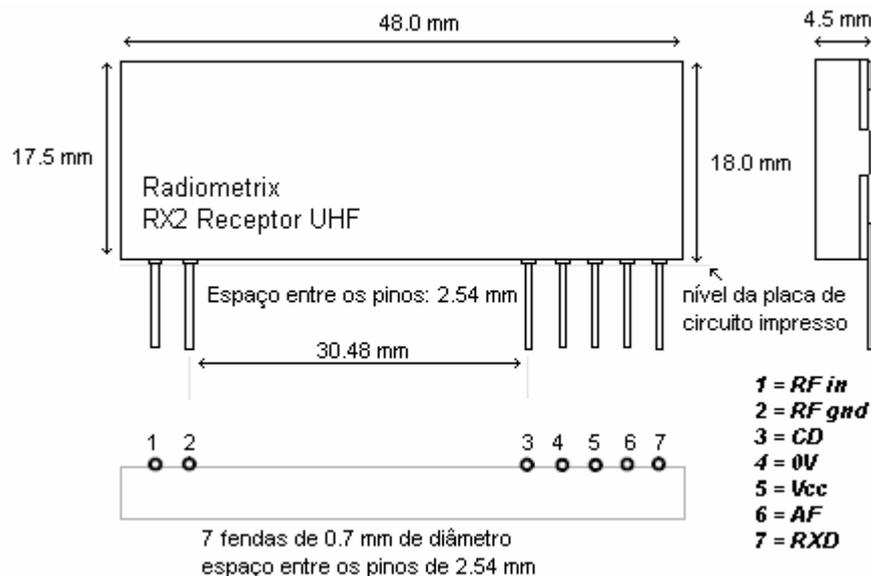


Figura 3.4: Dimensões físicas do receptor [13].

A tabela 3.2, possui as características e funções de cada um dos pinos do receptor.

Tabela 3.2: Descrição da função dos pinos do receptor [13].

Número do Pino	Nome	Descrição
1	RF IN	Entrada RF de antena de 50Ω. É isolada internamente da corrente.
2	RF GND	Pino do terra de RF. Internamente ligada ao pino 4 (0V).
3	CD	Pode ser utilizado para se obter sinal em níveis lógicos de <i>Carrier Detect</i> . Para isso deve ser usado um transistor PNP. Caso não seja necessário, este pino deve ser ligado diretamente ao pino 5.
4	0V	Terra do módulo de recepção
5	Vcc	Alimentação do módulo de recepção
6	AF	É a saída filtrada do sinal analógico do demodulador de FM. É utilizado como ponto de teste ou para saída de dados para decodificadores lineares
7	RXD	O sinal digital proveniente deste pino é uma versão em onda quadrada do sinal do pino 6. Pode ser utilizado para enviar informações para decodificadores externos. Os dados que saem deste pino são dados fiéis aos dados alimentados no transmissor. A impedância deste pino é $>1k\Omega$ e $< 1nF$.

Receptores são dispositivos eletrônicos que selecionam, amplificam e demodulam sinais de radiofrequência captados por uma antena [9]. O sinal externo recebido pelo pino 1 (RF IN), é enviado e tratado pela etapa de rádio frequência, composto por um filtro passa-banda de 418MHz, um pré-amplificador e um filtro passa-banda tipo SAW [13]. Neste estágio, o objetivo do circuito de amplificação de rádio frequência é aumentar a sensibilidade e o ganho do receptor [9].

O sinal passa por um conversor de frequência, composto por um misturador ligado a um oscilador local controlado por SAW a 418MHz. Em seguida, o sinal passa para a etapa de amplificação de frequência intermediária (FI), onde o sinal irá ser novamente filtrado dos sinais de frequência próximos ao do sinal desejado. Outra característica desta etapa é o ganho do sinal. O amplificador FI é responsável pela amplificação de alta frequência antes da demodulação do sinal e é composto por um segundo misturador de frequência ligado a um oscilador local a 15,82MHz e pelo demodulador de sinal [9].

Finalmente, o sinal passa por um novo filtro passa-baixa de 91kHz de frequência e por um buffer. Dois caminhos podem ser percorridos após o tratamento: o caminho que leva ao pino 6 (AF), onde há a saída do sinal analógico, ou o caminho do pino 7 (DATA OUT), que passa por um adaptador (*adaptive data slicer*), que fornecerá a saída do sinal em sua forma digital [13].

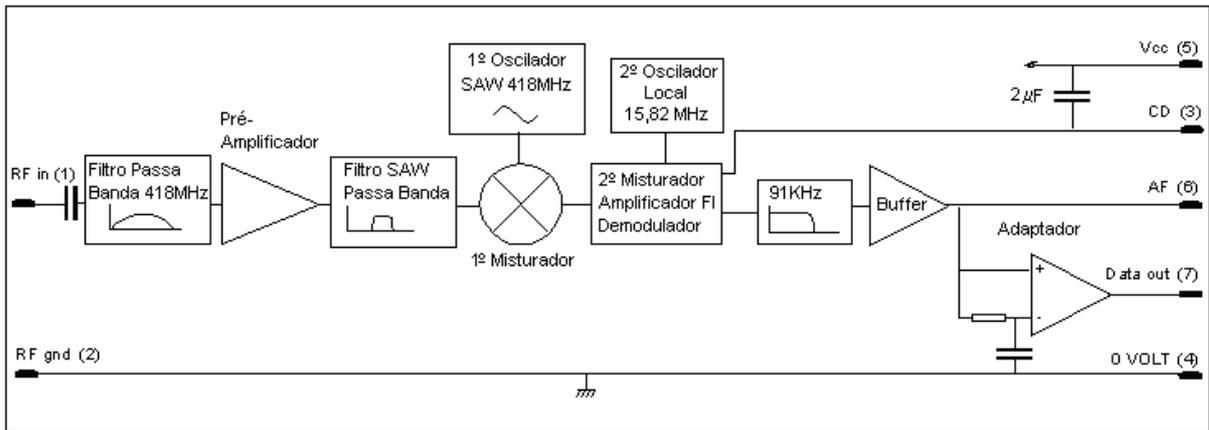


Figura 3.5: Diagrama de blocos de funcionamento do receptor [13].

A implementação do receptor (figura 3.6) é um pouco menos simples que a implementação do transmissor, pois existem duas configurações disponíveis, uma delas contendo um transistor PNP, para fazer a separação da informação do pino 3 (CD) para sistemas que necessitem deste dado. Caso o CD não seja necessário, o pino 3 é ligado diretamente ao pino 5 (Vcc), como está representado na figura.

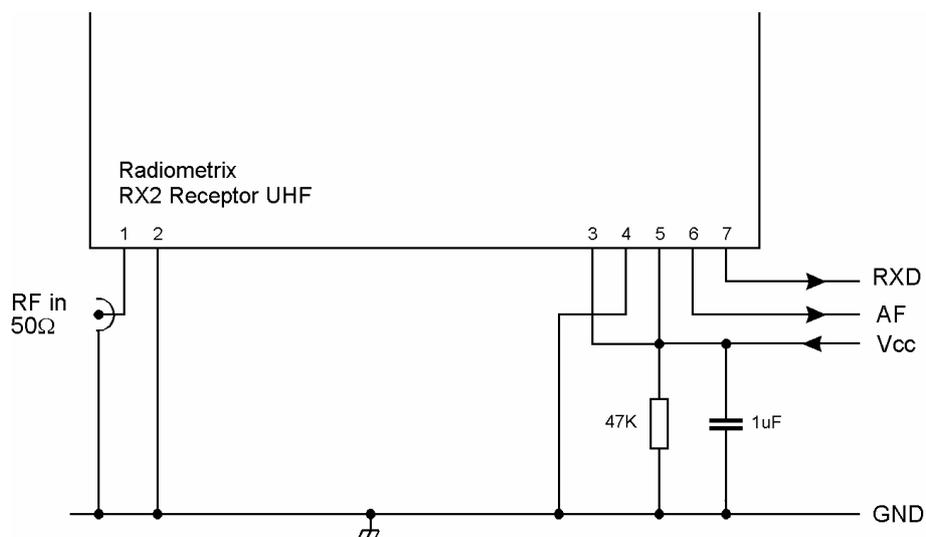


Figura 3.6: Circuito padrão do receptor [13].

3.2.3 A Antena

O alcance de um sinal de rádio frequência pode variar muito de acordo com o tipo de antena e do meio em que é utilizado. Escolher o melhor design, e que combine com as necessidades de um projeto, é uma tarefa complexa que requer diversos testes com equipamentos sofisticados e demanda conhecimentos avançados de propagação de ondas de rádio frequência. A escolha da antena errada pode prejudicar o sucesso do projeto [6, 7].

Alguns cuidados devem ser tomados para o funcionamento ótimo de uma antena. Qual é a proximidade de outros objetos condutores da antena? Metais, partes do corpo humano, como as mãos e pés, fontes de energia tais como transformadores, pilhas, baterias e etc, são geradores de interferência e podem prejudicar a performance de uma antena. Por esta razão, dependendo do modelo escolhido, o posicionamento da antena deve ser cuidadosamente escolhido e deve estar mais longe possível de outros condutores elétricos [6, 7].

O fabricante dos módulos de comunicação recomenda, e aprova, três tipos de antena. São elas do tipo helicoidal, do tipo loop e do tipo chicote ou haste. Cada uma delas tem características diferentes entre si como a performance, facilidade de implementação, tamanho, imunidade a ruídos gerados por outros objetos e o alcance. Na tabela a seguir existe uma classificação dos tipos de antena, quanto aos quesitos citados anteriormente [13].

Tabela 3.3: Avaliação dos tipos de antena [6, 7,13].

	Helicoidal	Loop	Chicote
Performance	Boa	Regular	Ótimo
Facilidade de implementação	Boa	Regular	Ótimo
Tamanho	Ótimo	Bom	Regular
Imunidade a efeitos de proximidade	Boa	Ótimo	Regular
Alcance em ambiente aberto	200m	100m	300m

A antena helicoidal (figura 3.7) pode ser um fio de cobre em espiras, diretamente conectado ao pino 2 do módulo transmissor e/ou no pino 1 do receptor. Essa antena é muito eficiente dado o seu tamanho reduzido (20,0mm x 4,0mm de diâmetro). Porém, sua proximidade a outros objetos condutores reduz sua eficiência drasticamente. Possui fácil implementação, devido à facilidade de encontrar os materiais necessários para a sua

construção, entretanto as espiras devem estar no tamanho correto para não prejudicar ainda mais a sua performance. Possui a vantagem de ser uma antena pequena e discreta, ótima para sistemas portáteis. Seu alcance é satisfatório, mas não atinge o máximo que os módulos de comunicação podem oferecer [6, 7].



Figura 3.7: Antena helicoidal [6, 7].

Já a antena loop (figura 3.8) nada mais é que uma trilha em torno de uma placa de circuito impresso. Em seu início e no final da volta, estão presentes capacitores comuns e capacitores variáveis para dar o ajuste fino ao sinal [13]. Tem alta imunidade a proximidade de sinais que podem causar interferência, no entanto, apresenta a pior performance de todas as antenas e sua implementação requer recursos difíceis de se obter. Recomendada apenas para sistemas que utilizarão placas de circuito impresso e que não requerem distâncias muito longas (abaixo de 100m) [6 e 7].



Figura 3.8: Antena tipo loop [6, 7].

A antena tipo chicote (figura 3.9) pode ser um fio de cobre, uma haste, uma trilha em uma placa de circuito impresso ou uma combinação das três opções. É conectada diretamente ao módulo e seu tamanho total ideal está em torno de 16,0 cm (1/4 do comprimento da onda a uma frequência de 418MHz) [13].



Figura 3.9: Antenas tipo chicote [6, 7].

Seu desempenho é excelente, podendo alcançar os 300m suportados pelos módulos de comunicação. No entanto, a antena deve ser afastada de qualquer material metálico a fim de prevenir sérios problemas com efeitos de interferência. É o tipo de antena que ocupa o maior tamanho entre os tipos citados, o que pode ser uma desvantagem em sistemas portáteis [6, 7].

Portanto, apresentados os diversos tipos de antena, a tarefa de escolher o tipo que atenda às necessidades do projeto se torna mais fácil. Como o projeto que foi desenvolvido necessita de portabilidade, fácil implementação, um desempenho satisfatório e que tenha certa imunidade a ruídos, foi utilizada a antena do tipo helicoidal. Contudo, uma outra ótima escolha é a antena do tipo chicote, pois é a que melhor atende aos itens de alcance de transmissão.

Respeitando a sugestão do fabricante dos módulos de comunicação, a antena helicoidal deve ser feita utilizando um fio de cobre encapado de 0,5mm de diâmetro, possuir 24 espiras com diâmetro interno de aproximadamente 3,2mm e atingir uma altura de 20,0mm [13].

Capítulo 4

Resultados e Conclusões

Neste projeto foi desenvolvida uma comunicação digital de dados através de módulos de comunicação de rádio frequência para comando de um microcontrolador 8051. O desenvolvimento do trabalho exigiu conhecimentos de configuração e operação da porta serial de um microcomputador comum e de um microcontrolador da família 8051. Teorias de modulação de sinais, funcionamento de transmissores e receptores de FM, foram fundamentais para entender o funcionamento interno dos módulos de comunicação.

Foi desenvolvido o programa da estação base em linguagem C/C++ que proporcionou a configuração, conexão e envio de mensagens da porta serial do microcomputador para o módulo de transmissão. O programa desenvolvido pela estação remota foi escrito em linguagem Assembly e permitiu o funcionamento correto do tratamento do protocolo pelo microcontrolador e acionamento do periférico.

Foram vistos como são feitos a programação e o controle da porta serial em ambiente Win32 através da API do sistema operacional. Não foram desenvolvidos quaisquer estudos da aplicação deste projeto em ambiente Linux por não ser foco deste trabalho, mas é uma alternativa interessante e de maior flexibilidade, pois possui o seu código fonte aberto e não exige a utilização de programas intermediários para acesso a porta serial.

Embora a existência de algumas fontes de interferência tenham perturbado o bom funcionamento da transmissão em alguns momentos, os resultados de transmissão foram atingidos com eficácia, pois o periférico formado por uma seqüência de LEDs respondeu com sucesso aos comandos enviados pela estação base. Recomenda-se o estudo de maneiras de minimizar a presença destes ruídos ou o estudo do comportamento do sistema em presença destas frequências indesejadas.

Como sugestões para futuros trabalhos ficam as seguintes propostas:

- Comunicação full-duplex: os módulos de comunicação podem ser substituídos por um par de transceptores. Essa alteração permite que sejam enviadas e recebidas informações simultaneamente, permitindo que a estação base e a estação remota se comuniquem.

- Rede de microcontroladores 8051: Com a possibilidade de comunicação full-duplex, existe a possibilidade de criação de uma rede de microcontroladores controlados por uma ou mais estações base.
- Criação de uma estrutura avançada de protocolos: a possibilidade de uma rede de microcontroladores exige a criação de protocolos mais fortes e melhor estruturados.
- Utilização de Computadores de Mão: é uma alternativa ao microcomputador pessoal. Os módulos de comunicação foram desenhados visando também este tipo de aplicação (existem versões que funcionam a 5,0V e a 3,0V).
- Periféricos: o canal de controle foi apresentado neste projeto, mas existem inúmeras possibilidades de aplicação que vão de sistema de monitoramento remoto de processos industriais a controle de iluminação doméstica sem fio;

Anexo 1

Esquemas dos circuitos utilizados no projeto

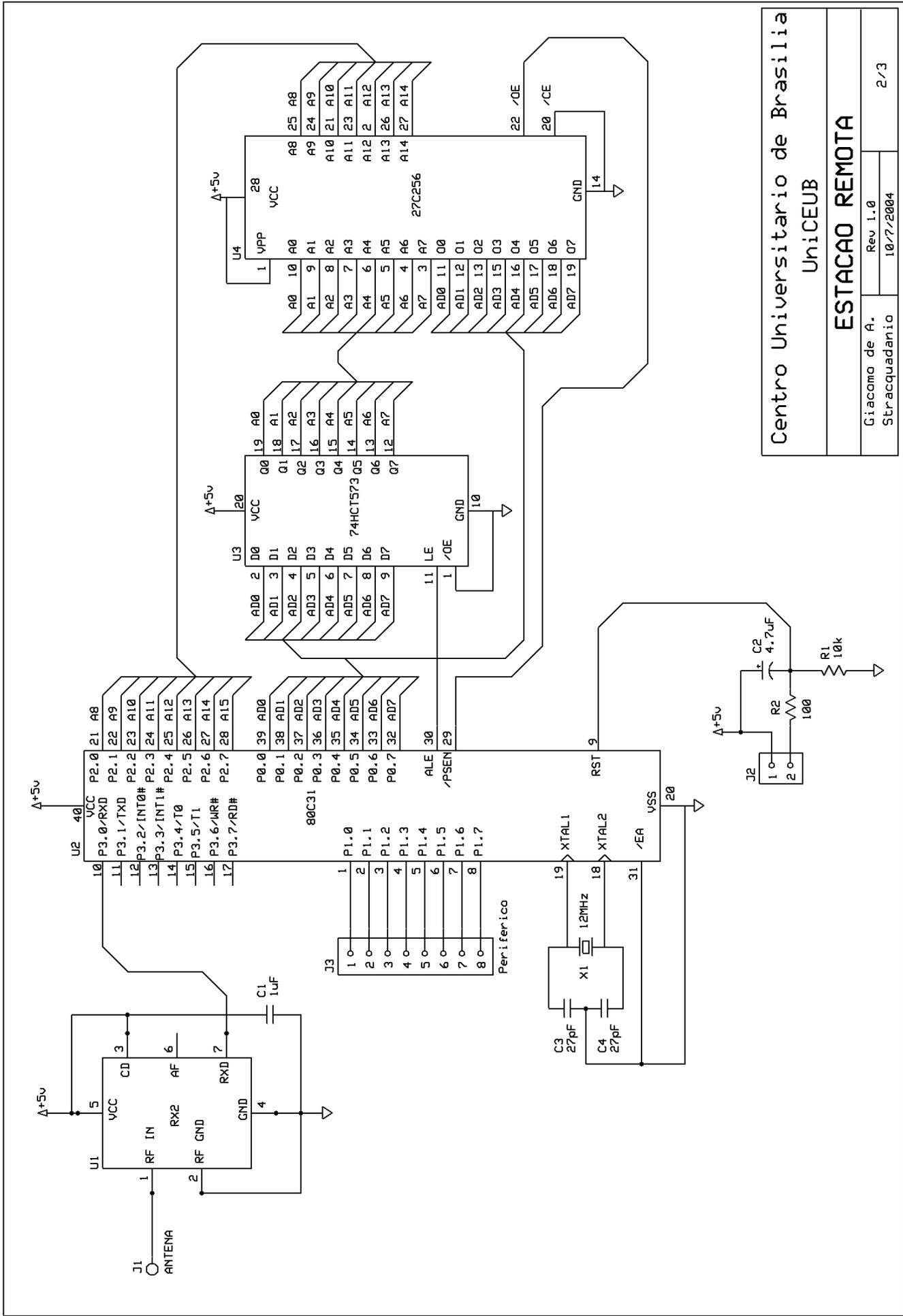
Os esquemas apresentados a seguir foram desenhados utilizando o software gratuito e de livre distribuição *ExpresSCH* versão 4.1.2. encontrado no site www.expresspcb.com.

O primeiro esquema representa a montagem da interface entre a estação base e o módulo de transmissão de dados TX2 da Radiometrix.

O segundo esquema apresenta a montagem da estação remota representados a sua ligação com o receptor RX2 da Radiometrix, como é feita a ligação com a memória EPROM, onde é gravado o programa da estação remota, o circuito de RESET e a sua saída para a conexão com o periférico.

O terceiro esquema é a representação do periférico utilizado neste projeto. É a ligação simples de diversos LEDs conectados a transistores afim de melhorar a luminosidade.

Todos os valores de componentes como resistores e capacitores foram retirados dos seus respectivos *datasheets*.



Centro Universitario de Brasilia

UniCEUB

ESTACAO REMOTA

Giacomo de A. Stracquadanio	Rev 1.0	2/3
	10/7/2004	

Anexo 2

Programa em Linguagem Assembly da Estação Base

```
#####  
; Centro Universitário de Brasília - UniCEUB  
; Faculdade de Ciências Exatas e de Tecnologia - FAET  
; Engenharia de Computação  
; Projeto Final de Conclusão de Curso  
;  
; Comunicação Wireless: Uma Proposta de controle da família de  
; microcontroladores 8051.  
; Autor: Giacomo de Almeida Stracquadanio  
; Orientadora: Prof. MC. Maria Marony  
;  
; Data: 06/2004  
;  
; Programa de Recepção de Dados Via Porta Serial  
; Versão: Final  
;  
#####  
  
org    000h  
  
ljmp   030h  
  
org    023h  
  
ljmp   SERIAL  
  
#####  
;  
; Configuração da Porta Serial  
;  
#####  
  
org    030h  
  
CONFIG:  
  
mov    IE,#090h           ;Habilita a interrupção do canal serial. Libe-  
                          ;ra EA (Enable All) e ES (Enable Serial).  
mov    SCON,#050h        ;Move para SCON o valor 54h = 01010000b. Isso  
                          ;deixará a porta serial no modo 1 e habilitará  
                          ;a recepção.  
                          ;SM0  SM1  SM2  REN  TB8  RB8  TI  RI  
                          ; 0   1   0   1   0   0   0   0  
  
mov    PCON,#080h  
  
                          ;Configuração da taxa de transmissão  
  
mov    TMOD,#022h        ;Move para TMOD o valor 22h = 00100010b. Isso  
                          ;habilita o timer 0 e timer 1 no modo 2.  
mov    TH1,#0E6h         ;Como SMOD = 1 e TH1 = E6h, portanto a taxa de  
                          ;transferência fica em 2400 bps.  
mov    TL1,#0E6h
```

```

setb TR1                ;Inicia o Timer 1

ljmp 300h

;#####
;
;Tratamento do Protocolo
;
;#####

og 300h

SERIAL:

jnb RI,SERIAL           ;Aguarda recebimento de dados da porta serial
mov R1,SBUF             ;Transfere para o registrador o dado recebido
                        ;pela porta serial

clr RI                  ;Habilita a interrupção serial a receber novos
                        ;dados

ret                     ;Retorna da chamada da sub-rotina

;-----

Inicio_Pacote:

call SERIAL             ;Chama a sub-rotina SERIAL

cjne A,#04Ch,Inicio_Pacote ;Caso o valor recebido seja diferente de 04Ch,
                        ;retorna a subrotina Inicio_Pacote. Se o valor
                        ;recebido for igual ao valor de inicio
                        ;de pacote então executa o comando seguinte

jmp Endereco_Remota    ;Chama a sub-rotina Endereco_Remota

;-----

Endereco_Remota:

call SERIAL             ;Chama a sub-rotina SERIAL

cjne A,#077h,Inicio_Pacote ;Caso o valor recebido seja diferente de 077h,
                        ;retorna a subrotina Inicio_Pacote. Se o valor
                        ;recebido for igual ao valor de inicio
                        ;de pacote então executa o comando seguinte

jmp Endereco_Base      ;Chama a sub-rotina Endereco_Remota

;-----

Endereco_Base:

call SERIAL             ;Chama a sub-rotina SERIAL

cjne A,#051h,Inicio_Pacote ;Caso o valor recebido seja diferente de 051h,
                        ;retorna a subrotina Inicio_Pacote. Se o valor
                        ;recebido for igual ao valor de inicio

```

```

;de pacote então executa o comando seguinte

jmp  ACENDE_LED_PAR          ;Chama a sub-rotina ACENDE_LED_PAR

;#####
;
;Programação da Sequencia de Acende e Pisca Led
;
;#####

ACENDE_LED_PAR:

cjne  R1,#02Ah,ACENDE_LED_IMPAR
mov   P1,#055h
jmp   Inicio_Pacote

;-----

ACENDE_LED_IMPAR:

cjne  R1,#03Bh,ACENDE_LED_TODOS
mov   P1,#0AAh
jmp   Inicio_Pacote

;-----

ACENDE_LED_TODOS:

cjne  R1,#04Ch,PISCA_LED_PAR
mov   P1,#0FFh
jmp   Inicio_Pacote

;-----

PISCA_LED_PAR:

cjne  R1,#042h,PISCA_LED_IMPAR
mov   R2,#000h
jmp   PISCA_PAR

PISCA_PAR:

mov   P1,#055h
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
mov   P1,#000h
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
inc   R2
cjne  R2,#005h,PISCA_PAR
ljmp  Inicio_Pacote

```

```

;-----
VOLTAR:

ljmp Inicio_Pacote

;-----

PISCA_LED_IMPAR:

cjne R1,#06Eh,PISCA_LED_TODOS
mov R2,#000h
jmp PISCA_IMPAR

PISCA_IMPAR:

mov P1,#0AAh
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
mov P1,#000h
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
inc R2
cjne R2,#005h,PISCA_IMPAR
ljmp Inicio_Pacote

;-----

PISCA_LED_TODOS:

cjne R1,#050h,VOLTAR
mov R2,#00h
jmp PISCA_TODOS

PISCA_TODOS:

mov P1,#0FFh
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
mov P1,#000h
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
lcall DELAY
inc R2
cjne R2,#005h,PISCA_TODOS
ljmp Inicio_Pacote

```

```

;#####
;
;Configuração do Timer 0 (Delay)
;
;#####

org    3DFh

DELAY:

mov    TH0,#high(65535-15535)
mov    TL0,#low(65535-15535)
setb   TR0
jnb    TF0,$
mov    B,R0
inc    B
mov    R0,B
clr    TR0
clr    TF0
cjne   R0,#0FFh,DELAY
ret
end

```

Bibliografia

- [1] BORLAND SOFTWARE CORP. **Borland C++ Help**. Versão 6. 2002
- [2] CAMPBELL, Colin K. **Surface Acoustic Wave Devices for Mobile and Wireless Communications**. Academic Press: Boston, 1998.
- [3] DENVER, Allen. **Serial Communications in Win32**. MSDN Library, 1995. Disponível em: <http://msdn.microsoft.com/library/en-us/dnfiles/html/msdn_serial.asp>. Acesso em: 13 abr. 2004.
- [4] DORMAN, Andy. **Wireless Communication: O Guia Essencial de Comunicação Sem Fio**. Rio de Janeiro: Editora Campus, 2001.
- [5] LATHI B. P. **Modern Digital and Analog Communication Systems**. 3rd edition. New York: Oxford University Press, 1998.
- [6] LINX TECHNOLOGIES, INC. **RM Series Receiver Module Data Guide**. Grants Pass, 2000.
- [7] LINX TECHNOLOGIES, INC. **RM Series Transmitter Module Data Guide**. Grants Pass, 2000.
- [8] MATEUS, César Augusto. **C++ Builder 5: Guia Prático**. São Paulo: Editora Érica, 2000.
- [9] NASCIMENTO, Juarez. **Telecomunicações**. São Paulo: Makron Books, 2000.
- [10] NICOLOSI, Denys E. C. **Microcontrolador 8051 Detalhado**. São Paulo: Editora Érica, 2000.
- [11] NICOLOSI, Denys E. C. **Laboratório de Microcontroladores Família 8051: Treino de Instruções, Hardware e Software**. São Paulo: Editora Érica, 2002.

- [12] PEACOCK, Craig. **Interfacing the Serial/RS232 Port**. 2001. Disponível em <<http://beyondlogic.org/serial/serial.htm>>. Acesso em 21 mar. 2004
- [13] RADIOMETRIX. **TX2 & RX2 Data Sheet**. Watford, 2002.
- [14] SCHILDT, Herbert. **Borland C++ Builder: Referência Completa**. Rio de Janeiro: Editora Campus, 2001.
- [15] SILVA JÚNIOR, Vidal Pereira da. **Aplicações Práticas do Microcontrolador 8051**. 11ª Edição: Com Novo Visual. São Paulo: Editora Érica, 2003.
- [16] TEXAS INSTRUMENTS, INC. **MAX 232, MAX 232I Dual EIA-232 Drivers/Receivers Data Sheet**, Dallas, 2004
- [17] VASCONCELOS, Laércio. **Hardware Total**. São Paulo: Makron Books, 2002.