Centro Universitário de Brasília – UniCEUB
Faculdade de Ciências Exatas e Tecnologia – FAET
Curso de Engenharia da Computação
Projeto Final

CONTROLE DO APARELHO DE TELEVISÃO VIA COMANDO DE VOZ UTILIZANDO MICROCONTROLADOR

Leonardo Lins de Albuquerque Lima

RA: 2006460-6

Professora Orientadora:

Profª. M.C. Maria Marony Sousa Farias Nascimento

Brasília - DF

Dezembro de 2007

LEONARDO LINS DE ALBUQUERQUE LIMA

CONTROLE DO APARELHO DE TELEVISÃO VIA COMANDO DE VOZ UTILIZANDO MICROCONTROLADOR

Monografia apresentada à banca examinadora para conclusão do curso e obtenção do título de bacharel em Engenharia de Computação do Centro Universitário de Brasília - UniCEUB.

Brasília – DF

Dezembro de 2007

AGRADECIMENTOS

Agradeço muito à minha família, por acreditar em mim, demonstrarem apoio irrestrito em todos os momentos e entenderem minha ausência em diversas ocasiões durante estes anos;

À professora Maria Marony Sousa Farias Nascimento, por sua competente orientação, interesse, dedicação e presteza, imprescindíveis para o incremento deste projeto;

A todos os professores que contribuíram para minha formação, em especial ao Coordenador Albiezer e ao professor Javier, pela atenção e profissionalismo, extremamente importantes na minha vida acadêmica e na realização deste trabalho;

Aos amigos que constantemente me incentivaram no desenvolvimento desta monografia, especialmente a Marcos Felipe, com quem compartilhei várias dúvidas e tive valiosas discussões, e a Leonardo de Paula e Silva, que nos momentos necessários muito contribuiu para a viabilização deste projeto;

Aos colegas e funcionários do UniCEUB, pelo companheirismo e pela amizade estabelecida.

"A imaginação é mais importante que a ciência, porque a ciência é limitada, ao passo que a imaginação abrange o mundo inteiro.”

Albert Einstein

RESUMO

Com a finalidade de beneficiar os usuários com dificuldades motoras e aqueles que buscam sofisticação e conforto, foi desenvolvido para os aparelhos de televisão um recurso de automação residencial por meio do comando de voz. Neste projeto, o usuário controla as operações básicas de um aparelho de televisão utilizando apenas a voz. No computador, o Formulário de Entrada de Dados recebe o comando informado oralmente e o transmite através da interface serial para o microcontrolador, que é o responsável pelo acionamento do circuito do controle remoto.

Palavras-chave: Automação Residencial, Comando de Voz, Microcontrolador, Linguagem de Programação Visual Basic, Linguagem de Programação Assembly.

ABSTRACT

With the proposal to benefits the users with habilities dificulties and others like needs confort and sofistication, was developed to the televisions equipments one residential automatation inovation with the use of voice comand. In this project, the user controls the basics operations of the televisions equipments using only the voice. In the computer, the Data Enter System receive the comand informed orally and transmit trough the serial interface to the PLC, with is the responsable to the remote control circuit action.

Key-words: Residential Automation, Voice Command, Microcontroller, Programming Language Visual Basic, Programming Language Assembly.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE TRECHOS DE CÓDIGO

LISTA DE ABREVIAÇÕES E SIGLAS

Aureside	Associação Brasileira de Automação Residencial.
CFTV	Circuito fechado de televisão.
COM	Porta serial utilizada no projeto.
VB	Linguagem de Programação Visual Basic
PIC 16F877	Micro controlador utilizado no projeto.
dB	Decibel. Unidade de medida do nível de intensidade do som.
DB-9	Conector serial, padrão fisico RS-232, com nove pinos.
DB-25	Conector serial, padrão fisico RS-232, com 25 pinos.
EPROM	Electrically Programmable Read-Only Memory (memória somente de leitura programável eletronicamente).
IDE	Integrated Development Environment (ambiente de desenvolvimento integrado).
Mbps	Megabits por segundo.
RAM	Random Access Memory (memória de acesso aleatório). Memória volátil, ou seja, perde os dados quando o computador é desligado.
RS-232	Padrão fisico da interface serial que especifica a quantidade de conectores e a tensão nos pinos.
UART	Universal Asynchronous Receiver and Transmitter (transmissor e receptor assíncrono universal).
USB	Universal Serial Bus (barramento serial universal).

CAPÍTULO 1. INTRODUÇÃO

Neste projeto foi desenvolvida uma solução de automação residencial através do comando de voz utilizando um microcontrolador que faz o elo entre o computador e a interface elétrica. Para isso, foram desenvolvidos softwares para o computador responsável pela interface com o usuário - e para o microcontrolador - responsável por controlar o circuito de acionamento elétrico do controle remoto da televisão.

No Capítulo 1 é feita uma introdução ao projeto, onde são apresentados os objetivos e motivações. Na seção 1.1 são apresentadas as motivações do autor para o desenvolvimento do projeto. Na seção 1.2 é mostrada uma visão geral do projeto e a forma de organização da monografia.

1.1 Motivação

O fato de a literatura técnica a respeito do assunto abordado ser escassa – o que pode ser evidenciado pela existência de apenas um trabalho relacionado à questão da automação residencial através de comando de voz no UniCEUB - e a crescente popularização do tema, foram as principais motivações para o desenvolvimento deste projeto. O tema abordado foi desenvolvido no intuito de promover real eficiência e originalidade no recurso apresentado e no método de implementação.

1.1.1 Objetivos gerais

O objetivo deste projeto é proporcionar maior facilidade e melhoria para a sociedade através do desenvolvimento de um tema voltado para a área de automação e controle. O avanço na aplicação da tecnologia aqui apresentada resulta da praticidade proporcionada pelo comando de voz que controla as funções básicas de um aparelho de televisão.

1.1.2 Objetivos específicos

Este trabalho pretende mostrar a validade da utilização de uma nova opção para os usuários de aparelhos televisivos, mais especificamente para os telespectadores que possuem algum tipo de dificuldade de locomoção, como os portadores de necessidades especiais, os idosos, assim como as pessoas com necessidades especiais temporárias, como as gestantes, ou mesmo para alguém que encontra comodidade em acionar o aparelho de televisão por comando de voz. Dessa forma, o usuário poderá controlar seus equipamentos sem a necessidade de se locomover ou solicitar a terceiros.

Os benefícios advindos da automação através de comandos de voz estendem-se também a todos os consumidores que não aspiram apenas a produtos, mas a conceitos, como conforto e modernidade e sofisticação.

1.2 Visão Geral do Projeto

A solução desenvolvida é mostrada na figura 1.1:

Figura 1.1 Diagrama Geral do Projeto

1) O usuário informa o comando no microfone.
2) O software de reconhecimento de voz traduz a informação, preenchendo um campo específico no Formulário de Entrada de Dados desenvolvido na linguagem de programação Visual Basic.

3) Através do software, é feita a validação do comando, a) se for válido, o computador envia um bit respectivo ao comando para o microcontrolador b) se não for válido, desconsidera e aguarda o próximo comando.
4) O usuário é informado sobre o comando válido.
5) O microcontrolador é responsável pelo acionamento do dispositivo infravermelho que altera o status no televisor.

Cada uma dessas etapas é detalhada nos capítulos da monografia, conforme divisão a seguir:

- Capítulo 1: Introdução, com a motivação para a escolha do tema.
- Capitulo 2: Material teórico sobre automação residencial e comando de voz.
- Capítulo 3: Exposição do hardware e interfaces gráficas desenvolvidas.
- Capítulo 4: Apresentação do formulário de entrada de dados, suas finalidades e seu funcionamento, software desenvolvido em Assembly para o microcontrolador, cuja função é a de interpretar os comandos solicitados e repassá-los para o dispositivo infravermelho.
- Capitulo 5: Apresentação estatística dos dados coletados durante a avaliação do desempenho e os resultados obtidos.
- Capítulo 6: Considerações finais do projeto, com as conclusões, dificuldades encontradas e sugestões para trabalhos futuros.
- Apêndices: Documentação dos códigos-fonte dos softwares de reconhecimento do comando e do microcontrolador.

CAPÍTULO 2. REFERENCIAL TEÓRICO

Neste capítulo é apresentada uma introdução teórica sobre os principais assuntos do projeto. Na seção 2.1 é abordada a automação residencial no Brasil com algumas características. Na seção 2.2 é apresentada a tecnologia de reconhecimento de voz e suas peculiaridades.

2.1 Automação Residencial

A automação via comando de voz une duas vertentes que estão em constante crescimento no mercado brasileiro. O conforto e a praticidade, e a tecnologia para obtenção destas, em pouco tempo será, indispensável.

O primeiro passo para entender a complexidade deste mercado é fixar-se no conceito de **integração de sistemas.** Isoladamente, cada um dos sistemas adotados numa residência tem a sua eficiência limitada. Utilizando-se o conceito de integração, o potencial de benefícios aumenta tremendamente. A operação fica mais simples, a economia e a segurança aumentam, o conforto estende-se pela casa toda [Aureside, 2007].

Surgiu depois de seus similares nas áreas industrial e comercial por motivos de escala de produção e econômicos, pois essas áreas propiciariam rapidez no retomo dos investimentos. No Brasil, os primeiros sistemas industriais automatizados surgiram na década de setenta. Depois de consolidada a automação industrial, o comércio entrou na era automatizada e os avanços da informática propiciaram inovações constantes [Aureside, 2006].

“O projetista de circuitos eletrônicos microcontrolados tem desempenhado um papel de destaque neste contexto, pois viabiliza o desenvolvimento de soluções personalizadas e de baixo custo. Uma exigência cada vez mais comum entre as empresas modernas. Este é um dos motivos que explica o extraordinário crescimento do uso de microcontroladores no projeto de circuitos eletrônicos, e um número cada vez maior de projetistas. Costuma-se dizer que o

limite de criação de soluções envolvendo microcontroladores está associado à criatividade do projetista. Quem é projetista sabe quanto dinheiro pode estar por trás de uma boa idéia." [ZANCO, 2006].

2.2 Reconhecimento de Voz

Existem dois parâmetros que classificam os sistemas de reconhecimento de voz. O primeiro parâmetro é a dependência ou não com relação ao falante. O segundo parâmetro é a forma da fala, discreta ou contínua [Jerome, 1994].

Um sistema independente reconhece a voz de qualquer usuário e não requer treinamento. Os sistemas dependentes são utilizados para reconhecimento de um número maior de vocabulários previamente treinados pelo usuário. Já os sistemas contínuos reconhecem a voz natural, enquanto nos sistemas discretos é indispensável que as palavras sejam pronunciadas pausadamente.

As primeiras soluções de automação residencial por meio de comandos de voz eram inovadoras e muito interessantes, mas faltavam confiabilidade e desempenho para que se tornassem efetivamente um método viável de controle. Com a evolução da informática, houve um aumento na capacidade de processamento associado à redução de custos, possibilitando dar continuidade aos projetos de automação residencial por comando de voz. Segundo a Associação Brasileira de Automação Residencial - Aureside -, testes mostram que esses projetos funcionam razoavelmente bem, mas é necessário que o microfone esteja próximo ao usuário para permitir um reconhecimento seguro. A maioria dos produtos requer treinamento por parte do usuário para criação do padrão de voz e, ainda assim, os resultados não são confiáveis quando submetidos a ruídos de ambiente, como barulho (sons) de aparelho condicionador de ar ou o ruído emitido por pessoas conversando. Existem alguns critérios que devem ser observados durante o desenvolvimento de um projeto de automação que utilize

comandos de voz [Aureside, 2006]:

- Confiabilidade;
- Possibilidade de operação em ambientes ruidosos e silenciosos;
- Dispensabilidade de utilização, por parte do usuário, de qualquer tipo adicional de hardware;
- Obrigatoriedade de uso de microfones distribuídos pela residência que captem todo tipo de som e reconheçam os comandos de voz;
- Atendimento do requisito de funcionamento paralelo com outros tipos de hardware, como interruptores e painéis de controle;
- Confirmação do comando recebido com opção de emissão sonora.

CAPÍTULO 3. HARDWARE E INTERFACES

Neste projeto existem três interfaces[1]. A primeira é a de entrada, que permite ao usuário informar o comando desejado. A segunda, é composta por dois softwares que trabalham em conjunto, o microcontrolador e o computador, que se comunicam por meio da interface serial. A terceira interface é a elétrica. Esta recebe os bits O ou 1 do microcontrolador e aciona o circuito do controle remoto, que por sua vez emite o infravermelho para a televisão.

Neste capítulo, são apresentadas as interfaces do projeto. Na seção 3.1 são mostrados os softwares que compõem a interface de entrada. Na seção 3.2 são tratados os detalhes da interface serial. Na seção 3.3 é exibida a interface elétrica do projeto.

3.1 Interface de Entrada

A interface de entrada é responsável pelo recebimento dos comandos do usuário, possibilitando o controle do aparelho de televisão. Neste projeto foi utilizado um computador para receber os comandos de voz do usuário através do Formulário de Entrada de Dados (ver seção 4.2). O usuário é capaz de manipular as funções básicas da televisão – aumentar/diminuir o som, passar/retornar o canal, ligar/desligar o aparelho - através do comando de voz, não havendo necessidade de digitação desses comandos.

Dois softwares trabalham em conjunto para possibilitar a entrada dos dados. Nos subitens 3.1.1 e 3.1.2 serão abordados o funcionamento do software de reconhecimento de voz e o funcionamento básico do Formulário de Entrada de Dados, respectivamente.

[1] Interface é a superfície que separa duas faces de um sistema. (Novo Dicionário Aurélio da Língua Portuguesa).

3.1.1 Reconhecimento de voz – IBM Via Voice

A proposta do projeto é utilizar o comando de voz para controlar o aparelho de televisão. O desenvolvimento de um sistema capaz de reconhecer padrões de vozes está fora do escopo deste trabalho e fica exposto como uma sugestão para projetos futuros.

Neste projeto, o reconhecimento de voz é realizado por um software proprietário da IBM, o Via Voice Pro USB Edition versão 9.0., que permite ao usuário ditar um texto, entre outras funções que não serão exploradas. Para que seja possível o reconhecimento da voz do usuário e para que os erros de interpretação por parte do software sejam minimizados, é necessária a criação de um padrão pessoal de voz, solicitada pelo software na primeira utilização ou na criação de um novo usuário. Para isso, é necessário que o usuário siga os passos do programa.

Ao ativar o IBM Via Voice, tudo o que o usuário disser no microfone será transcrito para o programa que estiver ativo no computador. No projeto, o programa ativo será o Formulário de Entrada de Dados.

A IBM disponibilizou, em setembro de 2004, sua tecnologia de reconhecimento de voz na forma de código aberto. A disponibilização de seu código fonte ocorreu após um acordo firmado com os maiores desenvolvedores de aplicativos de reconhecimento de voz, que se comprometeram a aderir ao desenvolvimento aberto e abandonar as soluções proprietárias.

3.1.2 Formulário de Entrada de Dados

O Formulário de Entrada de Dados opera em conjunto com o IBM Via Voice. Esse Formulário possui um campo de texto para recebimento dos comandos que são validados periodicamente. Dessa forma, o software recebe o comando ditado pelo usuário e transmite a informação para o microcontrolador. Os detalhes serão apresentados no Capítulo 4.

3.2 Interface Serial

As interfaces seriais estão nos computadores desde a década de oitenta. A principal característica das interfaces é que são capazes de transmitir ou receber um *bit* de cada vez.

É importante ressaltar que a transmissão serial no projeto será realizada utilizando-se os pinos de transmissão de dados (TX) pino 3 e recepção de dados (RX) pino 2, que possibilitarão o envio de dados entre o microcontrolador e o microcomputador. O aterramento, caso ocorra nível de tensão acima de 5 volts, é feito pelo pino 5.

Na Figura 3.1 é apresentada a configuração do cabo utilizado para efetivar a comunicação dos dados.

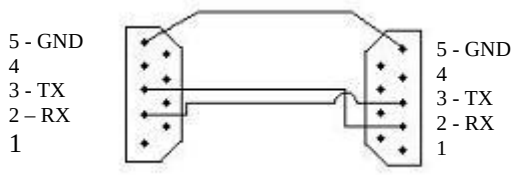


Figura 3.1 Configuração do cabo de comunicação serial utilizado

A comunicação é realizada com duas vias. A via de TX está ligada ao pino RC6 e a via de RX está ligada ao pino RC7 do microcontrolador, conforme pode ser observado na Tabela 3.1.

PIC	COM.
RC6	TX (saída)
RC7	RX (entrada)

Tabela 3.1 Pinagem no Microcontrolador PIC

A comunicação será implementada utilizando os recursos do próprio microcontrolador, via protocolo serial RS-232, que possibilitará a troca de mensagens entre o microcontrolador e o microcomputador.

Na Figura 3.2 podemos observar o conector RS232:

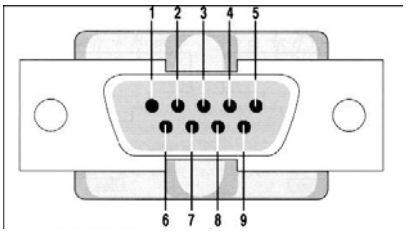


Figura 3.2 Conector RS232

Para que a integração dos componentes seja feita de forma íntegra e confiável, a taxa de transferência da comunicação será de 9.600 bps, tanto no microcontrolador quanto no microcomputador.

No microcomputador, a comunicação serial foi configurada com base na linguagem de programação em Visual Basic para recepção e tratamento dos dados. Por sua vez, no microcontrolador, a comunicação serial foi configurada em Assembly para recepção e envio de dados.

3.3 Kit Microcontrolador

O kit LABPIC, conjunto didático para desenvolvimento de projetos que emprega o microcontrolador PIC16F877, foi a opção utilizada, uma vez que esse kit possui diversos dispositivos integrados, como o display e a comunicação serial, o que facilitou a implementação deste projeto.

O kit LABPIC utilizado não possui ligações físicas permanentes entre o microcontrolador e os demais componentes do sistema, possibilitando a melhor adequação do projeto. [LABIT, 2007]

Na Figura 3.3 é mostrado o kit de desenvolvimento utilizado:

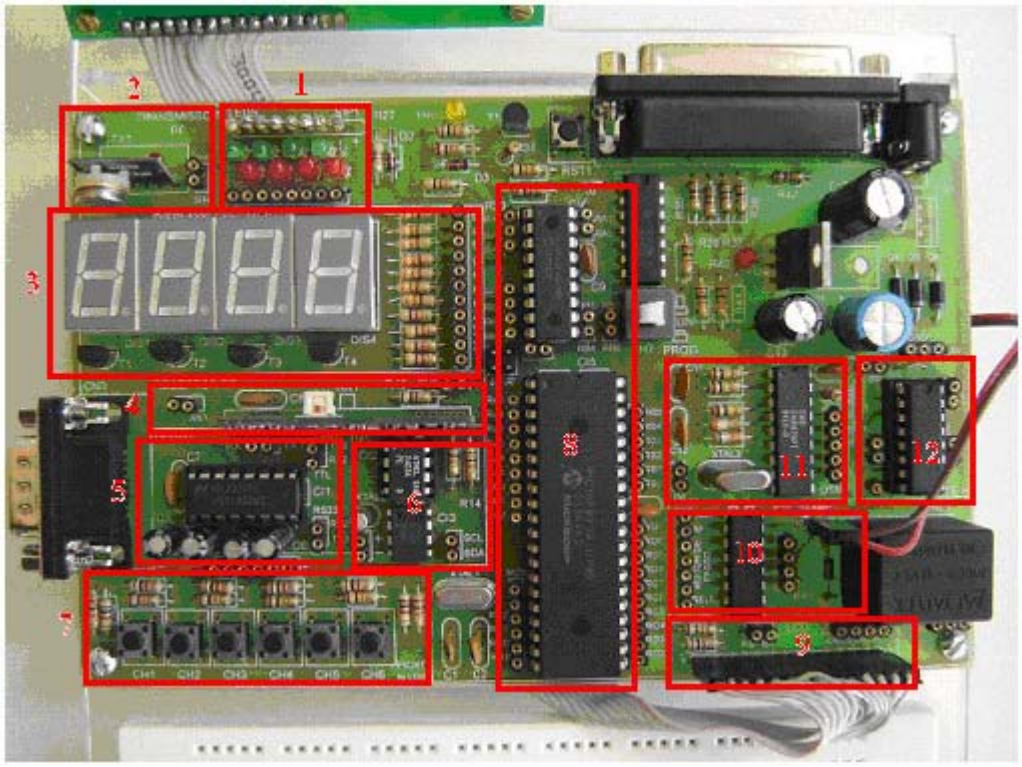


Figura 3.3 Kit de desenvolvimento LABPIC

O LABPIC é composto de 12 módulos:

1 – 8 leds, sendo quatro vermelhos e quatro verdes, anôdo comum, com ligações individuais;

2 – Transmissor RF 443MHz;

3 – 4 displays de 7 segmentos, catodo comum, com ligações individuais;

4 – Receptor RF 443MHz;

5 – Interface serial RS-232 e TTL;

6 – Bloco I2C, com memória EPROM 24C04 e DS1307;

7 – 6 chaves tipo push-button, com circuito fechado para o terra;

8 – 2 microcontroladores PIC: 16F877A e 16F628A, com um cristal de 4MHz, sendo que para fins deste trabalho será utilizado apenas o microcontrolador 16F877A;

9 – LCD ligado em 4 bits;

10 – Lâmpada, alto-falante, relé e driver para motor de passo;

11 – Decodificador DTMF;

12 – Ponte H;

Não obstante, serão empregados apenas os seguintes módulos: interface serial RS-232 e TTL; microcontrolador PIC 16F877A com um cristal de 4MHz; e LCD ligado em 4 bits;

Na Figura 3.4 são mostrados os pinos da porta paralela, que é utilizada para a gravação do código hexadecimal, desenvolvido em Assembly (ver seção 4.3) no PIC, através do kit LABPIC usado neste projeto:

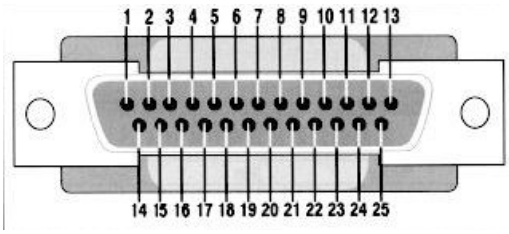


Figura 3.4 Conector da porta paralela

O código criado terá como exemplos as seguintes funções:

- Enviar os dados ao microcomputador pela porta serial;
- Verificar a resposta;
- Mostrar mensagem de status no display (ex.: Ligar!);
- Aviso sonoro;
- Em caso de comando aceito, acionar o respectivo botão no controle remoto;
- Receber resposta do microcontrolador;

3.4 Controle Remoto Infravermelho

O astrônomo William Herschel descobriu o infravermelho em 1800. William, sabendo que a luz solar continha todas as cores do espectro eletromagnético, queria saber qual a cor era a responsável pelo aquecimento dos objetos. Dessa forma, Herschel idealizou um experimento usando um prisma, papelão e termômetros com bulbos pretos, medindo as temperaturas de diferentes cores. Assim, concluiu que a temperatura mais elevada ocorria alem da luz vermelha e que não era visível ao olho humano. Na época, chamou essa radiação invisível de “raios caloríficos”, o que hoje é chamado de radiação infravermelha.

Existem vários tipos de detectores específicos para os mais variados tipos de aplicações infravermelhas.

Os detectores quânticos ou fótons detectores são dispositivos semicondutores cujas características elétricas são uma função do número de cargas que se tornaram disponíveis pela divisão de pares de elétrons causada pela colisão de fótons no material semicondutor. [Vanzetti Riccardo, 1972]

Para utilização deste dispositivo, foi empregado o controle remoto universal RCU403. A idéia de controlar o dispositivo via infravermelho serve para atender uma variedade de aparelhos que já estão no mercado com essa tecnologia disponível, sem descaracterizar o aparelho e facilitar a instalação nas residências.

Para transmitir dados, o sistema utiliza freqüências muito altas, um pouco abaixo da luz visível no espectro eletromagnético.

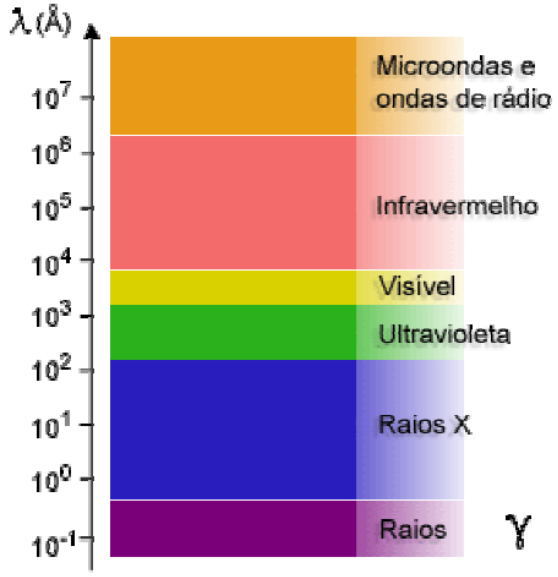


Figura 3.5 Espectro Eletromagnético (1 A = 10^{-8} cm)
Fonte: www.dimap.ufrn.br/~glaucia/RedesComputadores/8.MeiosdeTransmissao.pdf

Do mesmo modo que a luz, o sinal não pode penetrar em objetos opacos. Assim as transmissões por infravermelho são diretas ou difusas.

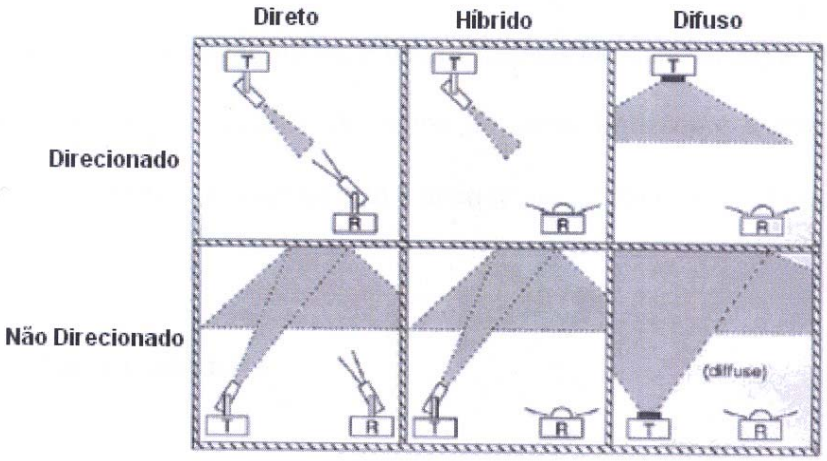


Figura 3.6 Formas de transmissão do infravermelho

Este trabalho é focado no modo difuso e não direcionado, ou seja, o campo de visão do receptor não precisa ser direcionado e restrito a uma pequena linha imaginária reta com o emissor. Ambos precisam estar no mesmo ambiente.

Figura 3.7 foto do controle remoto universal RCU403 adaptado para a utilização no trabalho.

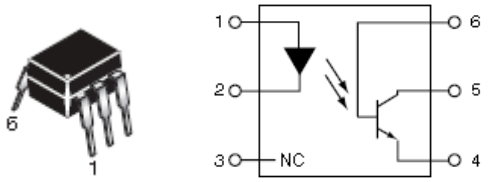
Figura 3.7 Controle remoto universal RCU403.

3.5 Descrição do Optoacoplador

A optoeletrônica combina elementos de tecnologia ótica e eletrônica. Os dispositivos optoeletrônicos que emitem ou detectam radiação ótica são denominados componentes optoeletrônicos. Os circuitos optoeletrônicos têm aplicações nas mais diversas áreas, tais como telecomunicações, controle e sensoriamento.

O optoacoplador é um componente eletrônico bastante utilizado em estruturas onde se deseja um isolamento total de sinal entre a entrada e a saída. Em diversas aplicações o terra da entrada não é o mesmo terra da saída. Daí a necessidade de uso de optoacopladores.

Na figura 3.5 é mostrado o encapsulamento (a), o esquema interno (b) e a descrição dos pinos (c) do optoacoplador 4N25, modelo utilizado no projeto.



1. Anodo
2. Catodo
3. Sem Conexão
4. Emissor
5. Coletor
6. Base

(a) (b) (c)

Figura 3.8 Optoacoplador 4n25

3.6 Circuito Eletrônico - Acionamento do infravermelho

O circuito de interface elétrica é apresentado na Figura 3.6. É composto pelos circuitos do microcontrolador e do acionamento do controle remoto. Este circuito simula o apertar de um botão no controle remoto correspondente ao comando de voz, emitindo assim o sinal infravermelho ao televisor.

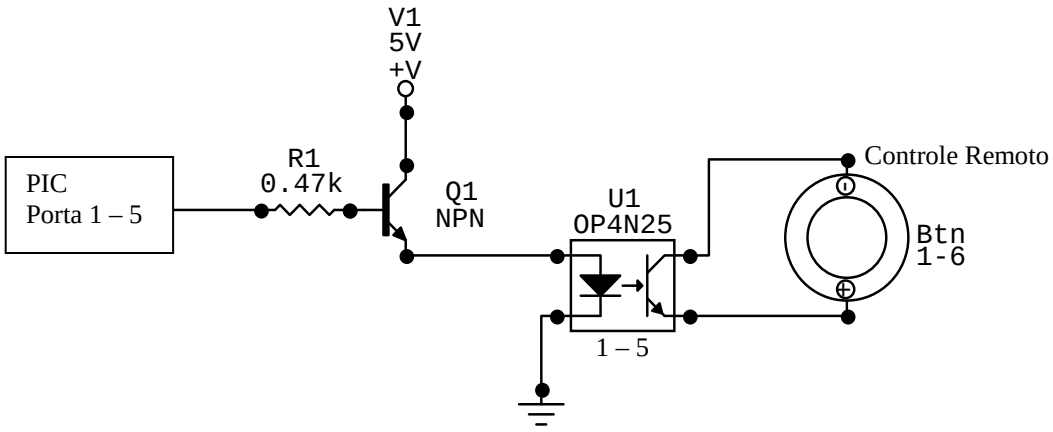


Figura 3.9 Circuito elétrico - Acionamento do infravermelho

Está exposto na Figura 3.9 o circuito de uma das portas do microcontrolador, sendo as outras cinco portas, cópias idênticas.

3.7 Desenvolvimento do Projeto Físico

Neste item é exibido parte do desenvolvimento físico do projeto. O desenvolvimento do trabalho no kit LabPic pode ser visualizado na Figura 3.10, e na Figura 3.11, o projeto concluído, onde são observados as ligações do LCD, da porta serial, dos cinco optoacopladores que acionam o infravermelho, do som para o aviso sonoro e do controle remoto.

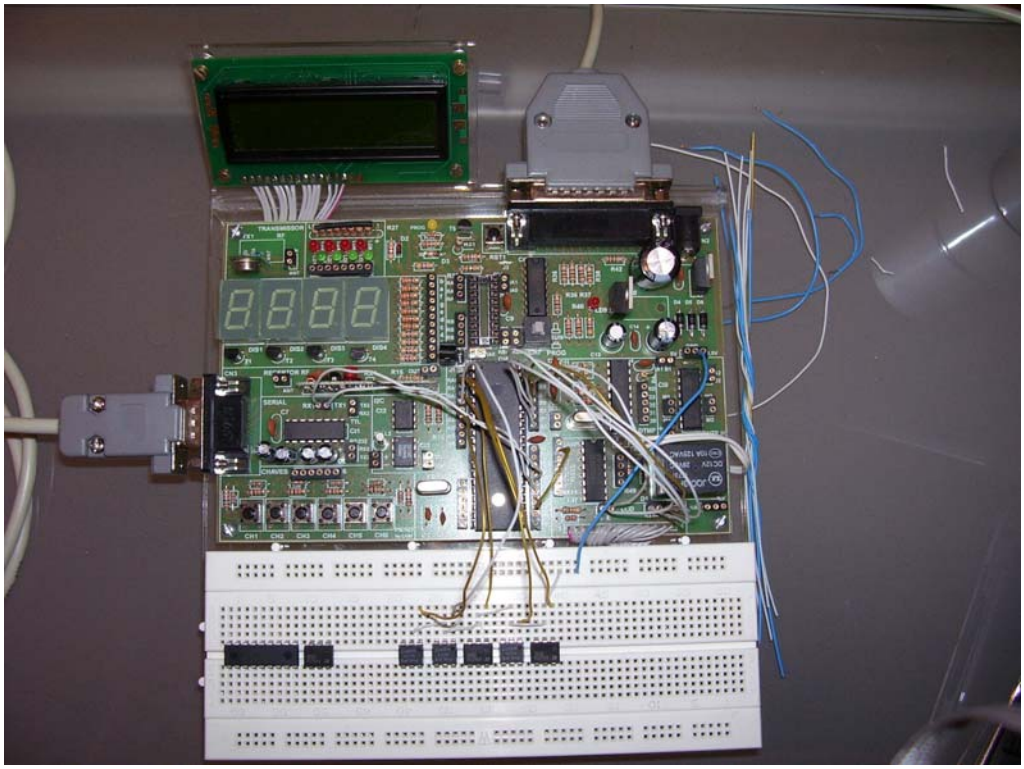


Figura 3.10 Projeto Físico - Desenvolvimento

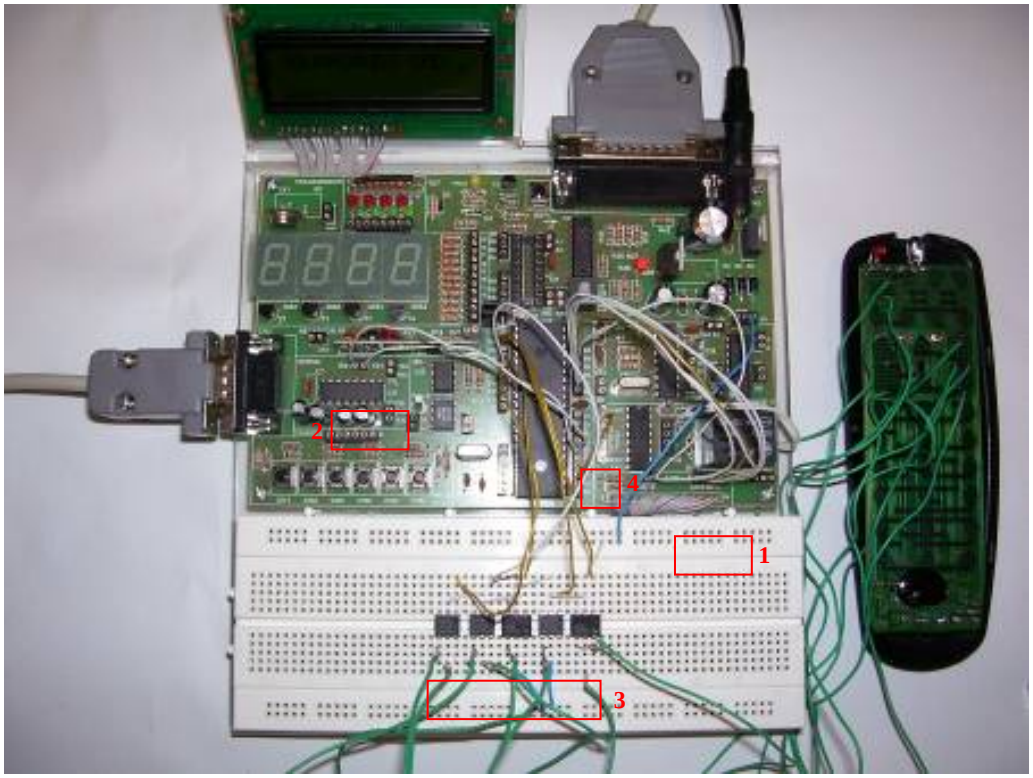


Figura 3.11 Projeto Físico - Completo

1 - Ligações do LCD;

2 - interface serial;

3 - Cinco optoacopladores;

4 - Som.

CAPÍTULO 4. SOFTWARES DESENVOLVIDOS

O computador recebe os comandos por meio de uma interface gráfica e envia ao microcontrolador, se os comandos forem válidos. O software do computador foi desenvolvido utilizando a linguagem Visual Basic da Microsoft. O microcontrolador opera o acionamento dos botões no controle remoto, e seu programa foi desenvolvido na linguagem Assembly.

A seção 4.1 trata sobre a linguagem de programação VB. Na seção 4.2 é explicado o funcionamento do Formulário de Entrada de Dados. Na seção 4.3 é realizada uma breve descrição do uso do Banco de Dados Access e a seção 4.4 analisa o código de programação Assembly.

4.1 Linguagem de Programação Visual Basic

O Visual Basic é uma linguagem de programação de autoria da empresa Microsoft, que integra o pacote Microsoft Visual Studio. Sua versão mais recente faz parte do pacote Visual Studio .Net, e é voltada para aplicações .Net. Sua versão anterior pertencia ao Microsoft Visual Studio 6.0, ainda muito utilizado atualmente.

Um aperfeiçoamento do Visual Basic é a linguagem que é dirigida por eventos (event driven) e possui também um ambiente de desenvolvimento integrado (IDE - Integrated Development Environment) totalmente gráfico, facilitanto de forma extraordinária a construção da interface das aplicações (GUI - Graphical User Interface), daí o nome "Visual".

Em suas primeiras versões, o Visual Basic não permitia acesso a bancos de dados, sendo portanto voltado apenas para iniciantes, mas devido ao sucesso entre as empresas - que faziam uso de componentes adicionais fabricados por terceiros para acesso a dados - a linguagem logo adotou tecnologias como DAO, RDO, e ADO, também da Microsoft, permitindo fácil acesso a bases de dados. Mais tarde foi adicionada também a possibilidade de criação de controles ActiveX, e, com a chegada do Visual Studio .NET, o Visual Basic

tornou-se uma linguagem totalmente orientada a objetos.

4.2 Formulário de Entrada de Dados

O Formulário de Entrada de Dados foi desenvolvido, como parte deste projeto, na Linguagem Visual Basic, fornecida pelo pacote Microsoft Visual Studio 6.0. Assim que o programa é inicializado, todos os comandos são salvos em uma tabela desenvolvida em Access, para levantamento estatístico posterior de palavras válidas e inválidas.

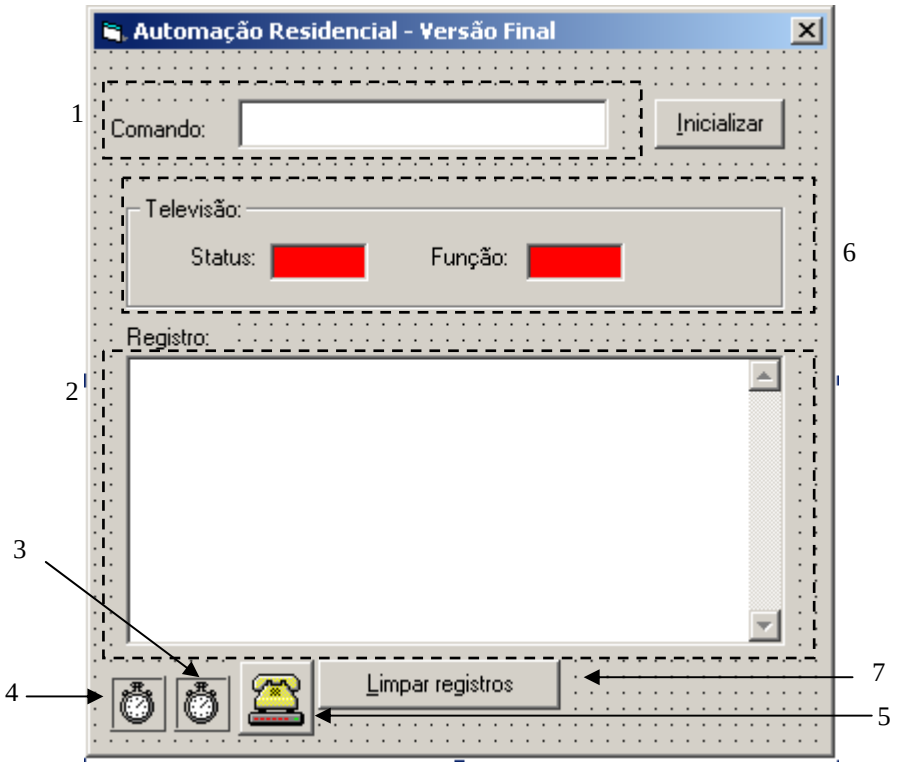


Figura 4.1 Tela de desenvolvimento do Formulário de Entrada de Dados

A Figura 4.1 foi dividida em trechos numerados para facilitar o entendimento:

1 – Campo de comandos, – recebe os comandos, faz a validação, caso seja um comando válido, envia um “bit” correspondente ao microcontrolador;

2 – Registro – Neste campo aparecerão diversas mensagens para o usuário, como, por exemplo, respostas aos comandos fornecidos.

3 – Controle do Temporizador – para validação dos comandos.

Assim que o programa é iniciado, é estabelecida a comunicação com o microcontrolador através da porta serial, retornando o texto “Porta Serial Inicializada” quando a conexão é realizada com sucesso. Neste projeto, a porta serial utilizada é a “COM1”. O comando é verificado automaticamente a cada segundo, limpando a caixa de texto em seguida.

4 – Controle do Temporizador para Controle das Funções – tempo padrão de 8 segundos por comando válido informado.

5 – Componente Microsoft Comm Control 6.0 – para configurar a interface serial de comunicação. (Baud Rate: 9600, sem paridade, DataBits: 8, Stopbit: 1)

6 – Status de entendimento do sistema e inicialização do temporizador para controle das funções.

7- Limpa o quadro de registro.

A Figura 4.2 mostra a tela do formulário após a inicialização

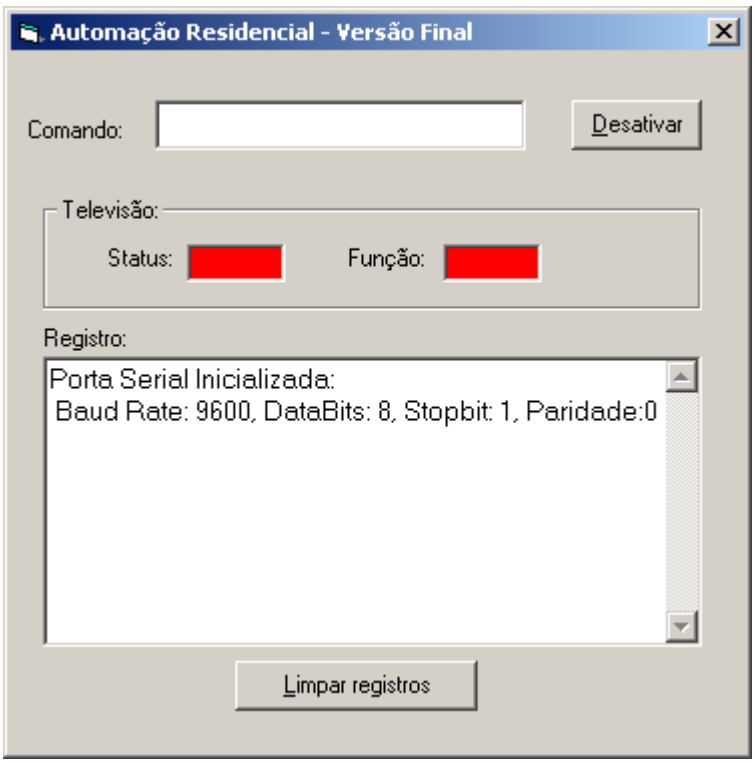


Figura 4.2 Sistema inicializado e conexão estabelecida

- O botão Inicializar foi substituído pelo Desativar

- Os componentes ficam invisíveis
- No quadro de registro consta informação sobre o status da conexão
- O temporizador para validação dos comandos é inicializado
- O campo comando recebe o foco do cursor

Na Figura 4.3 é apresentado o recebimento do comando “televisão”.

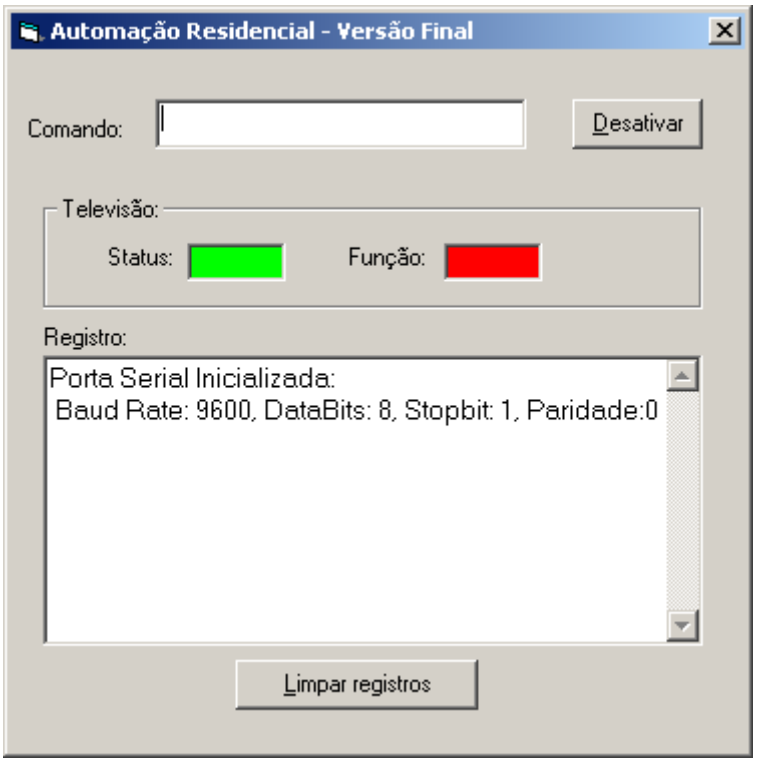


Figura 4.3 Recebimento do comando de controle do Objeto

- A Barra de status muda da cor vermelha para verde
- O Temporizador para Controle das Funções é iniciado – o sistema aguarda um comando válido por até 8 segundos
- Emissão de um aviso sonoro

Na Figura 4.4 é mostrado o status da função pela palavra “volume”, que é um comando válido

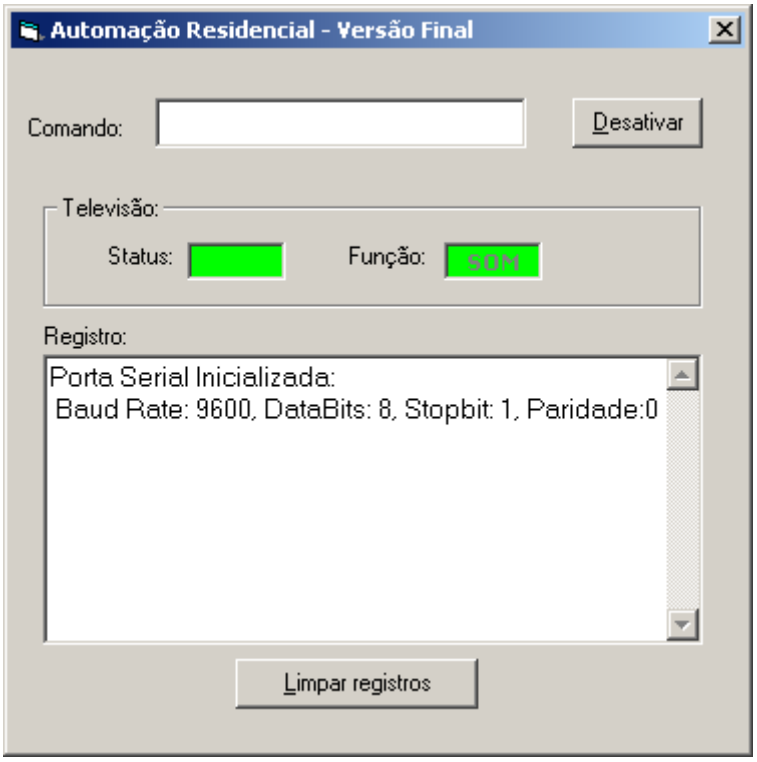


Figura 4.4 Recebimento do comando para controle da função

- A barra de função muda da cor vermelha para a cor verde
- Fica escrito dentro da barra de função o que será operado, no caso “SOM”
- Emissão de um aviso sonoro
- O Temporizador para Controle das Funções é reiniciado, com isso o usuário tem mais 8 segundos para informar qual a operação, ou mudar de função através do comando “canal”.

Na Figura 4.5 é apresentado o recebimento do comando “aumentar”.

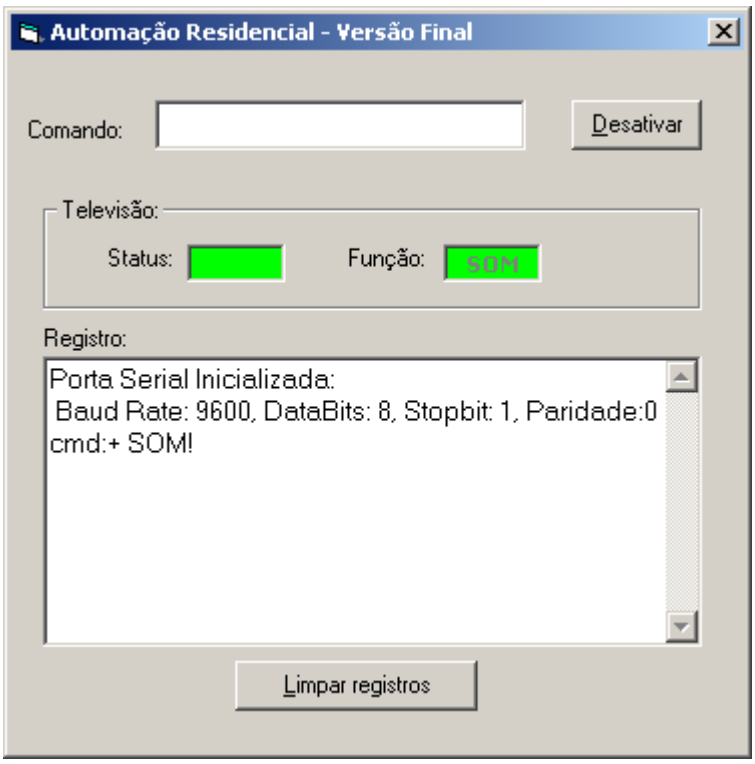


Figura 4.5 Operação de aumentar o som

- Com a entrada do comando válido, fica registrado no histórico o comando que foi enviado à porta serial.
    - O Temporizador para Controle das Funções é reiniciado; com isso o usuário tem mais 8 segundos para informar qual a operação, ou mudar de função através do comando “canal”.

Uma vez que o tempo de Controle das Funções é ultrapassado, é necessário o reinício da operação por meio do comando “televisão” ou similares, conforme lista da tabela no item 4.4 deste Capítulo.

4.3 Descrição do código fonte

Para facilitar o entendimento do funcionamento do software do computador, esta seção apresenta uma descrição de alguns trechos do código fonte dos módulos desenvolvidos em Visual Basic. Nos apêndices A, B e C são apresentados os códigos fontes completos.

4.3.1. Inicialização do sistema.

```
Private Sub Form_Load()
   '* Inicio do programa
   'Limpa variável de historico que alimenta o quadro de registro
   Historico = ""
   'Chamada para a função Limpar_Variaveis de controle
   Limpar_Variaveis
   'Constante de novo paragrafo no quadro de registro
   Paragrafo = Chr(13) & Chr(10)
   'Torna o botão Desativar invisivel
   btnDesativar.Visible = False
   'Torna o botão Inicializar visivel
   btnInicializar.Visible = True
   'Inicializa controle do loop de tempo
   Timer1.Enabled = False
   'Coloca o loop da controle de tempo em 2.5 segundos
   Timer1.Interval = 2500
‘Coloca o loop do segundo contador em 8 segundos
   Timer2.Interval = 8500
End Sub

Public Sub Limpar_Variaveis()
   Principal.Timer2.Enabled = False
   Cmd1 = ""
   Cmd2 = ""
   Cmd3 = ""
   varTV = False
   varCANAL = False
   varSOM = False
   Principal.Online.BackColor = &HFF&
   Principal.Online2.BackColor = &HFF&
   Principal.Online2.Text = ""
End Sub
```

Trecho de Código 4.1 Inicialização do sistema

Define o status inicial das variáveis utilizadas no formulário de entrada de dados e o tempo em cada temporizador.

4.3.2 Separação das palavras

Por meio desta rotina, o sistema tornou-se mais dinâmico e apresentou resultados mais precisos, mesmo estando em um ambiente com muito ruído.

A rotina foi desenvolvida com a finalidade de identificar as palavras mesmo que elas estejam contidas numa frase. Por exemplo: “Dados indicam aumento do volume de televisão negociado”, a rotina exposta no Trecho de Código 4.2 faz a separação das palavras de uma

frase e testa cada uma individualmente, ou seja, cada palavra na frase, inclusive as vogais são enviadas para comparação com o banco de dados. Para aumentar a precisão, todas as palavras são analisadas sem acentuação e em maiúsculo. [rotina: Retira_Acento(Palavra), Apêndice C]

```
-- Módulo Serial.bas--
Public Sub Verifica_Comando(txtCMD As String)
...
txtCMD = UCase(txtCMD) & " "     'função para deixar todo otexto em Maiúsculo
Do While Len(txtCMD) > 0 And (Left(txtCMD, 1) < "0" Or Left(txtCMD, 1) > "9")
        If InStr(" ", Left(txtCMD, 1)) = 0 Then
           Palavra = Palavra & Left(txtCMD, 1)
        Else
           If Len(Palavra) >= 1 Then
              Palavra = Retira_Acento(Palavra)
              Ordena_Comando Palavra
              Banco.Insere_Banco Palavra
           End If
           Palavra = ""
        End If
        txtCMD = Mid(txtCMD, 2)
     Loop
  ...
end sub
```

Trecho de Código 4.2 Separação das palavras

4.3.3 Ordenação dos comandos

Os comandos a serem executados, como: “Ligar a TV” ou “Aumentar o volume da televisão” são ordenados nas variáveis Cmd1, Cmd2 e Cmd3, para que a execução aconteça de forma ordenada, respeitando a ordem dos comandos: (i) o objeto a ser controlado, (ii) a função e (iii) como alterar. No caso do segundo exemplo, as variáveis são representadas da seguinte maneira: Cmd1 = televisão; Cmd2 = volume; Cmd3 = aumentar. O resultado dessa combinação é enviado ao microcontrolador, correspondendo a Tabela 4.3.

```
---  Chamada da Função Ordena_Comando, Módulo.Serial.bas ---
Private Sub Ordena_Comando(Palavra As String)
   If (Banco.Comando(Palavra, 8) = True) Then
      Cmd1 = Palavra
End If
   If (Banco.Comando(Palavra, 1) = True) Then
      Cmd2 = Palavra
      TpCmd2 = 1
   End If
   If (Banco.Comando(Palavra, 2) = True) Then
      Cmd2 = Palavra
      TpCmd2 = 2
```

```
    End If
    If (Banco.Comando(Palavra, 3) = True) Then
       Cmd2 = Palavra
       TpCmd2 = 3
    End If
    If (Banco.Comando(Palavra, 4) = True) Then
       Cmd2 = Palavra
       TpCmd2 = 4
    End If
    If (Banco.Comando(Palavra, 5) = True) Then
       Cmd3 = Palavra
       TpCmd3 = 5
    End If
    If (Banco.Comando(Palavra, 6) = True) Then
       Cmd3 = Palavra
       TpCmd3 = 6
    End If
End Sub

---  Função localizada no módulo Banco.bas ---
Public Function Comando(Palavra As String, Tipo As Integer) As Boolean
    Dim SQL As String
    Comando_liga = False
    SQL = "SELECT Comando FROM Comandos WHERE ID_Tipo = " & Tipo
    'Seta a variável bd como nova intancia de conexão
    Set rs = New ADODB.Recordset
    rs.Open SQL, strcon, adOpenStatic, adLockReadOnly
    Do While Not rs.EOF
       If rs("Comando") = Palavra Then
          Comando_liga = True
          Exit Do
       Else
          rs.MoveNext
       End If
    Loop
End Function
```

Trecho de Código 4.3 Ordenação dos comandos

Cada palavra é testada pela função “Comando” que recebe dois parâmetros, a palavra e o tipo da palavra. Se o teste for verdadeiro, ou seja, se a palavra for encontrada no banco de dados, é retornado o status de verdadeiro à função, que passa a executar as linhas na respectiva condição para posterior execução.

4.4 Banco de dados em Access

Uma base de dados simples foi criada neste projeto, a fim de facilitar a manipulação das informações. Contudo, o SGBD utilizado permite consultas e/ou acessos, como acontece em sistemas empresariais reais que possuem bancos de dados como Oracle, SQL ou DB2.

A função do banco de dados neste projeto é facilitar o levantamento de todos os comandos recebidos pelo Formulário de Entrada e minimizar os erros. Para tanto, foi utilizado o software Microsoft Access, que suporta a linguagem SQL, tendo em vista que este é um banco de dados simples e acessível, de fácil configuração e simples acesso por aplicações cliente-servidor. A Figura 4.6 demonstra a tabela desenvolvida para coleta de dados. Código fonte: BANCO.bas em anexo.

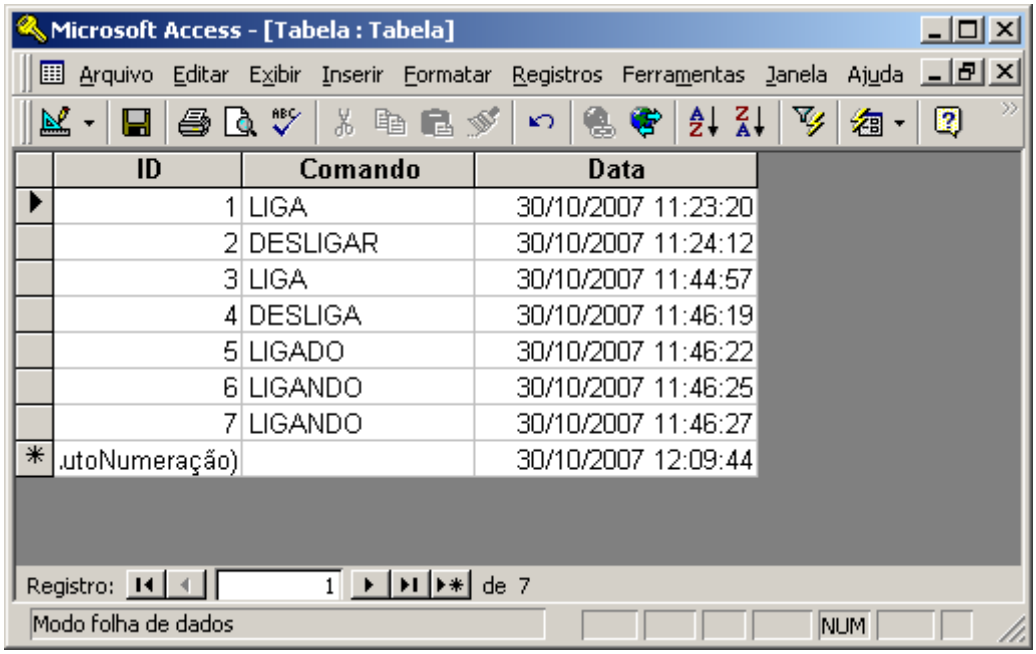


Figura 4.6 Tabela do Banco de Dados Access

Na tabela 4.1 são listados os tipos de comandos aceitos pelo sistema.

Tipo	
ID_Tipo	**Tipo**
1	LIGAR
2	DESLIGAR
3	CANAL
4	SOM
5	AUMENTAR
6	DIMINUIR
8	OBJETO

Tabela 4.1 Tipos de comandos

Para facilitar o estudo e estabelecer uma integridade das informações, foi criado um relacionamento entre as duas tabelas utilizadas no projeto. A tabela “Tipo” é relacionada com a tabela “Comandos”, respeitando o relacionamento de um para vários (1:N) com propagação de inclusão ou exclusão e atualização das informações contidas.

Na Figura 4.7 é exposto o relacionamento das duas tabelas desenvolvidas, para coleta de dados e posterior estudo, visando o aperfeiçoamento do sistema.

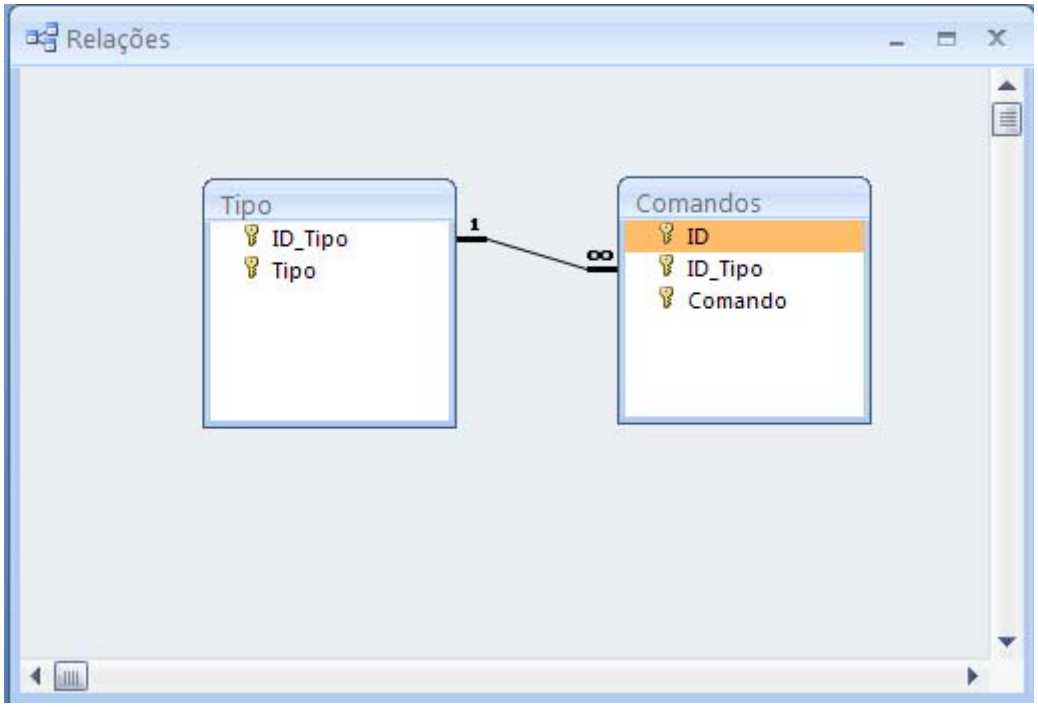


Figura 4.7 Relacionamento entre as tabelas do Access

Na tabela 4.2 são listados todos os comandos válidos pelo sistema. Sempre que for necessário, pode-se incluir mais um comando válido cadastrando na tabela de “comandos”, especificando o tipo do comando (ID_Tipo) e a palavra para acionamento (Comando).

Comandos					
ID	ID_Tipo	Comando	ID	ID_Tipo	Comando
3	LIGAR	LIGAR	53	LIGAR	ATIVAM
4	LIGAR	LIGANDO	58	OBJETO	VISAO
5	LIGAR	LIGADO	59	OBJETO	TELE
6	LIGAR	LIGAS	63	OBJETO	ALUSAO
7	LIGAR	LIGA	64	OBJETO	TELEVISAO
8	LIGAR	INICIO	54	OBJETO	ADESAO
9	LIGAR	INICIAR	55	OBJETO	TEVE
10	LIGAR	ON	56	OBJETO	TELEVISOR
58	LIGAR	ATIVAM	57	OBJETO	REVISAO
59	LIGAR	ATIVAR	61	OBJETO	TV
68	LIGAR	INICIALIZAR	62	OBJETO	LESAO
69	LIGAR	INICIAR-SE	28	AUMENTAR	AUMENTAR
61	DESLIGAR	FINALIZAR	29	AUMENTAR	AUMENTA
62	DESLIGAR	FINALIZAM	30	AUMENTAR	AUMENTANDO
11	DESLIGAR	DESLIGAR	31	AUMENTAR	ALTO
12	DESLIGAR	DESLIGANDO	32	AUMENTAR	+
13	DESLIGAR	DESLIGADO	33	AUMENTAR	AUMENTAM
14	DESLIGAR	DESLIGA	34	AUMENTAR	AUMENTE
15	DESLIGAR	FECHAR	35	AUMENTAR	SOBE
16	DESLIGAR	FECHA	36	AUMENTAR	SUBIR
64	DESLIGAR	ENCERRAR	37	AUMENTAR	MAIS
65	DESLIGAR	ENCERRAVA	38	AUMENTAR	MAS
66	DESLIGAR	ENCERRAR-SE	39	AUMENTAR	AUMENTO
67	DESLIGAR	ENCERRA	50	AUMENTAR	AMAR
17	CANAL	CANAL	60	AUMENTAR	ARMAR
18	CANAL	CANAIS	51	DIMINUIR	DESSE
19	CANAL	IMAGEM	40	DIMINUIR	DIMINUIR
20	CANAL	CANA	41	DIMINUIR	DIMINUINDO
21	SOM	SOM	42	DIMINUIR	DESCE
63	SOM	ANAL	43	DIMINUIR	BAIXO
22	SOM	VOLUME	44	DIMINUIR	DECREMENTAR
23	SOM	SONS	45	DIMINUIR	-
24	SOM	VOLUMES	46	DIMINUIR	MENOS
25	SOM	BARULHO	47	DIMINUIR	-DIMINUIR
26	SOM	VOL	48	DIMINUIR	DESCER
27	SOM	VOLUTIL			
52	SOM	SAO			

Tabela 4.2 Relação dos comandos válidos

Na tabela 4.3 são listados todos os possíveis caracteres que podem ser enviados pela porta serial do computador ao microcontrolador e sua respectiva funcionalidade.

CARACTER	FUNÇAO
1	LIGAR
2	DESLIGAR
3	SUBIR O CANAL
4	DESCER O CANAL
5	AUMENTAR O VOLUME
6	DIMINUIR O VOLUME

Tabela 4.3 Caracteres enviados pela porta serial e suas funcionalidades

4.5 Linguagem de Programação Assembly

Linguagem de montagem ou assembly é uma notação legível por humanos para o código de máquina utilizada por uma arquitetura de computador específica. A linguagem de máquina, que é um mero padrão de bits, torna-se legível pela substituição dos valores brutos por símbolos chamados mnemônicos.

A seguir são apresentadas as metodologias utilizadas para alcançar os objetivos (conforme relação exposta no Capítulo 3 item 3.3). Esses métodos estão de acordo com o código Assembly, apresentado integralmente em anexo.

Foram utilizadas as seguintes ferramentas para manipulação e compilação do código Assembly: (i) MpLab da Microsystem, para desenvolvimento do código em Assembly e compilação do código, gerando o hexadecimal a ser gravado no microcontrolador PIC; e (ii) IC-Prog, utilizado para gravar o código em hexadecimal na memória flash do PIC.

Ao final da compilação do Código, é criado um arquivo no diretório de trabalho com o

mesmo nome do programa fonte original, porém com extensão HEX. Esse arquivo será gravado no microcontrolador PIC, através do programa IC-PROG. [IC-PROG, 2007].

4.5.1 Inicialização do sistema e rotina principal

```
CBLOCK 20H
        SALVA_W     ; Salva conteúdo de W nas interrupções
        SALVA_S     ; Salva conteúdo de STATUS nas interrupções
        TEMPO1      ; Usado nas rotinas de tempo
        TEMPO2      ; Usado nas rotinas de tempo
        CHAR  ; Caracter ou comando a ser enviado p/ LCD
        DADO  ; Caracter recebido via RS-232
        TIMEOUTDB ; Caracter de validacao
        FREQ  ; Usado na rotina sonora
ENDC

ORG     000H
        GOTO  INICIO

; ***************** Tratamento das interrupções ******************

ORG     004H
        MOVWF       SALVA_W             ; Salva contexto
        SWAPF STATUS,W
        MOVWF       SALVA_S

        MOVF  RCREG,W               ; Lê dado recebido e zera flags
        MOVWF       DADO

FIMINT:
        SWAPF SALVA_S,W
        MOVWF       STATUS
        SWAPF SALVA_W,F
        SWAPF SALVA_W,W
RETFIE

LACO_INICIAL
        MOVLW       01              ; Apaga display

CALL    SEND_CMD
        MOVLW       0x80    ;Endereca a DDRAM do LCD para linha 1 e coluna 1 do
display
        CALL  SEND_CMD
        MOVLW       'A'
        CALL  SEND_CHAR
        MOVLW       'G'
        CALL  SEND_CHAR
        MOVLW       'U'
        CALL  SEND_CHAR
        MOVLW       'A'
        CALL  SEND_CHAR
        MOVLW       'R'
        CALL  SEND_CHAR
        MOVLW       'D'
```

```
CALL   SEND_CHAR
MOVLW          'A'
CALL   SEND_CHAR
MOVLW          'N'
CALL   SEND_CHAR
MOVLW          'D'
CALL   SEND_CHAR
MOVLW          'O'
CALL   SEND_CHAR
MOVLW          ' '
CALL   SEND_CHAR
MOVLW          'C'
CALL   SEND_CHAR
MOVLW          'M'
CALL   SEND_CHAR
MOVLW          'D'
CALL   SEND_CHAR
MOVLW          '.'
CALL   SEND_CHAR
MOVLW          '.'
CALL   SEND_CHAR
```

Trecho de Código 4.4 Variáveis e interrupção do microcontrolador

No trecho 4.4 é apresentada a definição das variáveis, a rotina de interrupção no microcontrolador e o laço principal onde aparece a mensagem de “Aguardando cmd” aguardando comando, no LCD. Sempre que a porta do dispositivo serial recebe informações, o sistema interrompe imediatamente o que esta fazendo e vai para a rotina que se inicia no endereço de memória ORG 004H do microcontrolador, o endereço de interrupção é um valor pré-definido do PIC. A informação recebida pelo dispositivo RS-232 é repassada para a variável DADO. Em seguida, a mesma é testada pela rotina exposta no Trecho de Código 4.5.

```
RECEBE_SERIAL
        MOVLW   '1'                     ; coloca o valor de 1 na variavel W
        SUBWF   DADO,W      ; subtrai variavel DADO do Valor da variavel W work
        BTFSC   STATUS,Z        ; verifica se o status Z esta clear: se sim vai (goto) funcao
VALIDACAO_OK
        GOTO   VALIDACAO_LIGA

        MOVLW   '2'                     ; coloca o valor de 2 na variavel W
        SUBWF   DADO,W      ; subtrai variavel DADO do Valor da variavel W work
        BTFSC   STATUS,Z        ;
        GOTO   VALIDACAO_DESLIGA

        MOVLW   '3'                     ; coloca o valor de 3 na variavel W
        SUBWF   DADO,W      ; subtrai variavel DADO do Valor da variavel W work
        BTFSC   STATUS,Z        ;
```

```
		GOTO	VALIDACAO_SOBE_CANAL

		MOVLW	'4'			; coloca o valor de 3 na variavel W
		SUBWF	DADO,W		; subtrai variavel DADO do Valor da variavel W work
		BTFSC	STATUS,Z	;
		GOTO	VALIDACAO_DESCE_CANAL

		MOVLW	'5'			; coloca o valor de 3 na variavel W
		SUBWF	DADO,W		; subtrai variavel DADO do Valor da variavel W work
		BTFSC	STATUS,Z	;
		GOTO	VALIDACAO_MAIS_SOM

		MOVLW	'6'			; coloca o valor de 3 na variavel W
		SUBWF	DADO,W		; subtrai variavel DADO do Valor da variavel W work
		BTFSC	STATUS,Z	;
		GOTO	VALIDACAO_MENOS_SOM

		MOVLW	d'5'		; coloca o valor de 5 na variavel W
		SUBWF	TIMEOUTDB,W	; subtrai variavel TIMEOUTDB do Valor da
variavel W work
		BTFSC	STATUS,Z	; verifica se o status Z esta clear: se sim vai (goto)
funcao TENTE_DE_NOVO
		GOTO	RECEBE_SERIAL
		INCF	TIMEOUTDB,F	;incrementa o valor da variavel TIMEOUTDB

		GOTO	LACO_INICIAL
```

Trecho de Código 4.5 Rotina de interrupção e validação

Assim que a variável DADO é validada, o microcontrolador repassa um caracter “A” para o computador, informando que o comando foi aceito. Um aviso sonoro também é acionado no microcontrolador e o display de LCD mostra a função que será acionada.

CAPÍTULO 5. AVALIAÇÃO DE DESEMPENHO

A viabilidade do projeto depende, principalmente, da capacidade do software IBM Via Voice interpretar corretamente o que o usuário está dizendo.

Para iniciar a simulação, foi necessário criar um padrão pessoal de voz no software IBM Via Voice. Após essa etapa, o software está preparado para uso, porém apresenta algumas falhas no reconhecimento de voz. Isso ocorre devido ao software generalizar algumas palavras que não foram treinadas.

O software funciona por aproximação nesses casos. Por exemplo: ao ditar a palavra "som", o programa interpretou a voz como "são". Dessa forma, para o correto entendimento de certas palavras pelo IBM via voice, é fundamental o treinamento exclusivo dessas palavras.

Neste capítulo são abordados as formas de avaliação de desempenho e os resultados obtidos. Na seção 5.1 são apresentados os dados obtidos nas análises de confiabilidade antes da inserção da utilização das rotinas de separação das palavras. Exposto no capítulo 4. Na seção 5.2 são apresentados os resultados após a utilização da rotina de separação das palavras.

5.1 Resultados obtidos – antes da rotina de separação das palavras

Na Figura 5.1 são mostrados os dados coletados antes da inclusão da rotina de separação das palavras (exposta no Capítulo 4) em um ambiente silencioso. No total foram 30 amostras para cada palavra. "Televisão" foi o destaque das palavras, a que apresentou maior confiabilidade, alcançando 97% de acertos, ou seja, 29 entre os 30 comandos ditos foram bem interpretados.

Os comandos relacionados são aqueles que obtiveram mais acertos. Um comparativo com os comandos mais utilizados também está exposto. Por exemplo: Televisão / Tv. E no caso do comando desligar que obteve uma taxa baixa de acertos, foi comparado com o

comando finalizar que obteve maior precisão.

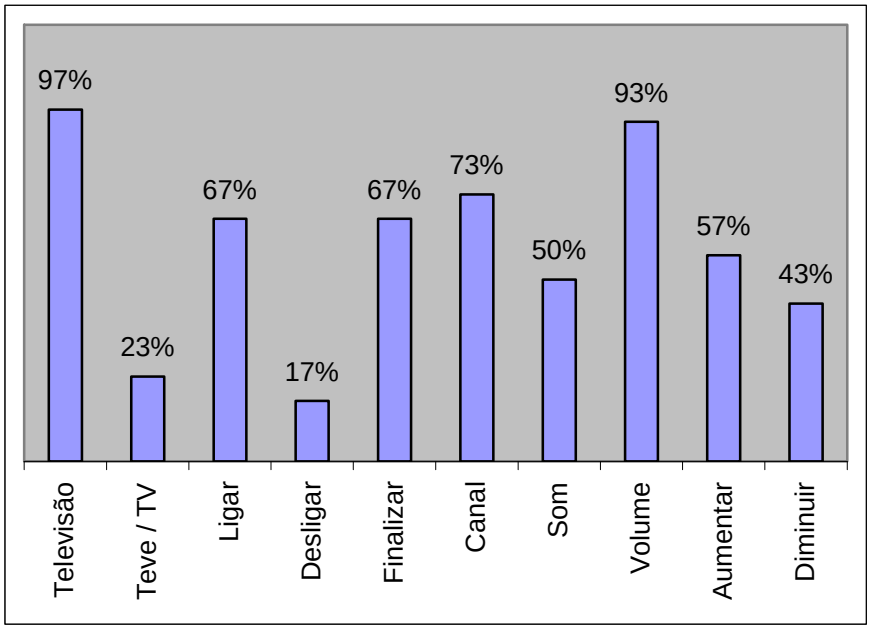


Figura 5.1 Estatística de acertos – sem a rotina de separação das palavras

Em um ambiente com qualquer tipo de ruído, onde o formulário de entrada de dados recebe uma grande quantidade de palavras, e sem a separação das mesmas, torna este projeto inviável, uma vez que a validação é feita a apenas uma palavra por vez.

5.2 Resultados obtidos – após a rotina de separação das palavras

Conforme exposto no Capítulo 4 itens 4.3.2 e 4.3.3, os testes foram considerados excelentes em ambientes silenciosos. Em vez de utilizar como critério a pronúncia das palavras individualmente, foi realizado um estudo sobre os acertos dos Tipo de Comandos do projeto. A Figura 5.2 mostra os resultados obtidos nos testes da validação quanto aos Tipos de Comandos, que foram apresentados no capítulo 4, Tabela 4.1 – Tipos de Comandos.

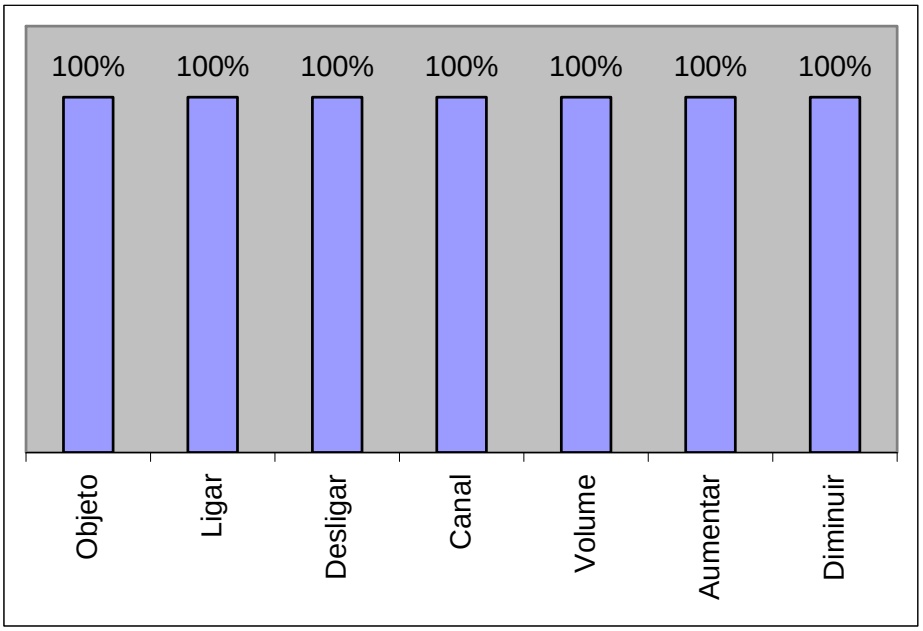


Figura 5.2 Estatística de acertos – Tipos de Comandos

Em ambientes com muita interferência sonora, de acordo com a Tabela 5.1, os resultados obtidos em uma simulação com aproximadamente 70 db de ruído, foram alterados, mas não prejudicam o funcionamento do projeto.

Na Tabela 5.1 são apresentados os valores aproximados do nível de ruído de alguns ambientes e objetos [Bertulani, 2007].

Nível de Intensidade (dB)	Intensidade do som (W/m^2)	Exemplos típicos
130	10	Limiar da percepção
120	1,0	Grande avião a jato
110	0,1	Grande orquestra
100	0,01	Arrebitamento
90	10^{-5}	Trem
80	10^{-4}	Escritório ruidoso
70	10^{-5}	Motor de carro
60	10^{-6}	Discurso
50	10^{-7}	Escritório com ruído médio
40	10^{-8}	Escritório quieto
30	10^{-9}	Biblioteca
20	10^{-10}	Sussurro
10	10^{-11}	Sussurro bem baixo
0	10^{-12}	Limiar da audibilidade (a 1000 Hz)

Tabela 5.1 Valores aproximados do nível de ruído

A única exigência para não diminuir o resultado do projeto em ambientes com muito ruído, é manter o microfone próximo ao usuário.

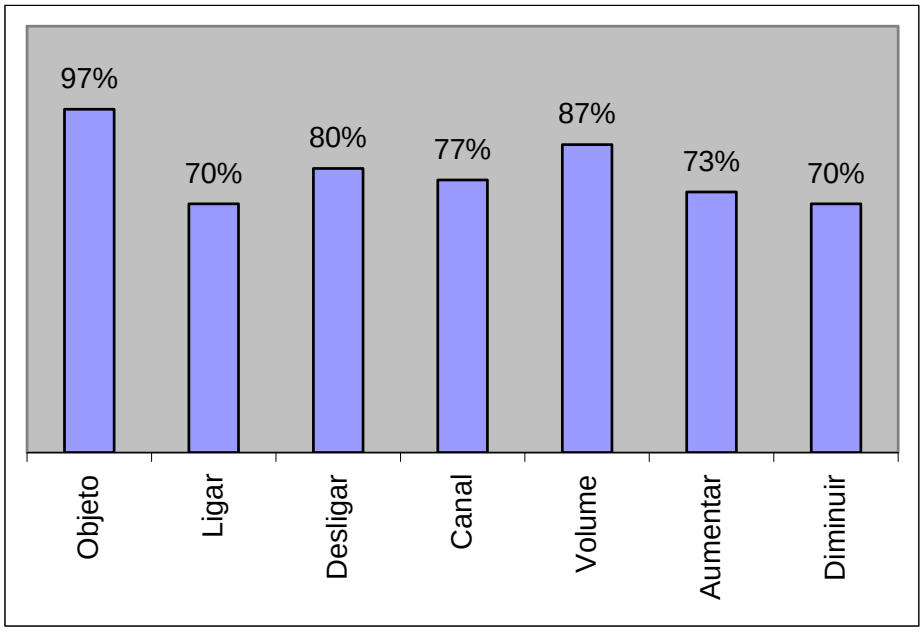


Figura 5.2 Estatística de acertos – ruído 70 dB – Tipos de comandos

Tabela 5.2 apresenta as palavras que são reconhecidas com maior precisão pelo padrão de voz desenvolvido para o projeto e que foram utilizadas nos testes desse projeto. O levantamento estatístico realizado simula um ambiente com aproximadamente 70 dB de ruído.

TIPO DO COMANDO	PALAVRA
OBJETO	TELEVISAO
LIGAR	INICIAR
DESLIGAR	ENCERRAR
CANAL	CANAL
SOM	VOLUME
AUMENTAR	AUMENTAR
DIMINUIR	MENOS

Tabela 5.2 Relação das palavras mais utilizadas

CAPÍTULO 6. CONSIDERAÇÕES FINAIS

Neste Capítulo são apresentadas as considerações finais relacionadas às etapas do projeto. Na seção 6.1 são apresentadas as conclusões. Na seção 6.2 são apontadas as dificuldades encontradas durante todas as fases do trabalho. Na seção 6.3 são apresentadas as sugestões para projetos futuros.

6.1 Conclusões

Neste trabalho foi apresentada uma proposta de automação residencial através de comando de voz. Durante o desenvolvimento do trabalho, foi observada uma gama de aplicações que ainda podem ser desenvolvidas dentro do mesmo tema..

Na fase de desenvolvimento do sistema, foi priorizada a interatividade com o usuário, e o resultado obtido mostrou ser eficiente e atendeu às expectativas, apesar das dificuldades encontradas em função da complexidade dos softwares desenvolvidos. Para alcançar esse resultado, cada etapa foi estabelecida, testes foram realizados e o desenvolvimento do projeto esteve sujeito a aplicação de conhecimentos adquiridos ao longo do curso, tais como: banco de dados, programação, lógica digital, microcontrolador e microprocessador.

Foi observada uma praticidade no controle de qualquer aparelho via comando de voz, o que oferece um aumento significativo na qualidade de vida do usuário, especialmente para a grande parcela da população que possui algum tipo de deficiência, portadores necessidades especiais permanentes, os idosos, as pessoas com algum tipo de necessidade especial temporária, como por exemplo as gestantes, alem de consumidores que buscam facilidade, conforto, e sofisticação.

Contudo, para alcançar a precisão adequada nos comandos, o software IBM Via Voice demanda muito tempo de dedicação ao treinamento do programa para o efetivo aperfeiçoamento do padrão pessoal de voz. O fato de ser um programa dependente – está

sujeito a treinamento para o aperfeiçoamento – e discreto – as palavras devem ser pronunciadas pausadamente –, essas especificações podem comprometer a confiabilidade do sistema na questão do programa ser suficientemente adequado para cumprir o que se propõe.

Alguns fatores, que são requisitos básicos para o desenvolvimento de soluções de automação residencial e para a tecnologia de comando de voz, foram atendidos neste projeto:

- Confiabilidade satisfatória;
- Interatividade com o usuário, proporcionada pelo Formulário de Entrada de Dados, o que facilita o usa da solução;
- Atualização tecnológica simples, uma vez que é utilizada uma linguagem de programação bastante difundida;
- Conforto e conveniência proporcionados pelo comando de voz;
- Possibilidade de integração de outros dispositivos e aparelhos domésticos;
- Status de confirmação do comando através do Formulário de Entrada de Dados e no microcontrolador;

6.2 Dificuldades encontradas

Diversas dificuldades foram encontradas durante a implementação do projeto. A primeira foi a falta de bibliografia específica sobre o tema.

Outra questão, foi a complexidade do código do Formulário de Entrada de Dados que, visando uma interação com o usuário, apresentou lentidão na leitura das rotinas e no acionamento do microcontrolador. A solução foi alterar a rotina de validação dos comandos e mudar a ferramenta de acesso ao banco de dados, tornando o andamento da resposta mais rápido.

Além disso, o acionamento da rotina de interrupção do pic demandou muito tempo. O fato de ser utilizado o Kit de desenvolvimento LabPic facilitou o entendimento e a

apresentação de progresso dos componentes físicos do projeto.

6.3 Sugestões para projetos futuros

A execução, o desenvolvimento e o emprego de outras aplicações, como por exemplo a automatização de uma casa que disponha o controle dos aparelhos eletrodomésticos utilizando receptor via rádio, como o dispositivo transmissor e receptor RF 443Mhz disponível para desenvolvimento no kit LabPic, conforme descrição no capitulo 3, item 3.3 kit de desenvolvimento.

Filtragem para atender à voz de apenas o usuário, na presença de ruídos ou conversa de outros. Desenvolvimento de uma central com chips de memórias acoplados, eliminando a necessidade do computado.

Desenvolvimento de um sistema de reconhecimento de voz utilizando, por exemplo, redes neurias.

REFERÊNCIAS

AHMED, Ashfaq. Eletrônica de Potência. São Paulo: Prentice Hall, 2000.

Atera Informática [Dome Page]. 2007. Disponível em: <http://www.atera.com.br>. Acesso em: 08 ago. 2007.

Aureside - Associação Brasileira de Automação Residencial [Dome Page]. 2007. Disponível em: <http://www.aureside.org.br>. Acesso em: 11 set. 2007.

BERTOLI, Roberto Angelo. Eletrônica. São Paulo: Colégio Técnico de Campinas, 2000.

BERTULANI, Carlos A. Ondas sonoras [Dome Page]. 2007. Disponível em: <http://www.if.uftj.br/teaching/fis2/ondas2/ondas2.htm1>. Acesso em: 15 nov.2007.

Fairchild Semiconductor Corporation. Datashett Optoacoplador 4N25. Jun. 2005.

HP [Dome Page]. 2007. Disponível em: <http://www.hp.com>. Acesso em: 13 nov. 2007.

JEROME, Jeffrey. Emerging Technologies for Independent Living - Report from a Working Conference. Maryland, 1994

LABIT. Manual do Kit de Desenvolvimento LABPIC – V.2.21. 2007

MALVINO, Albert Paul. Eletrônica: volume 1. 4. ed. São Paulo: Makron Books, 1995.

PCCOM [Dome Page]. 2007. Disponível em: <http://www.pccompci.com>. Acesso em: 22 out. 2007.

Rogercom [Dome Page]. 2007. Disponível em: <http://www.rogercom.com>. Acesso em: 11 out. 2007.

Sun Microsystems [Dome Page]. 2006. Disponível em: <http://www.sun.com>. Acesso em: 04 set. 2007.

Superdownloads - Circuit Maker Student Version [Dome Page]. 2007. Disponível em: <http://superdownloads.uo1.com.br/download/i9609.htm>. Acesso em: 14 out. 2007.

VASCONCELOS, Laércio. Dardware Total. São Paulo: Makron Books, 2002.

ZANCO, Wagner. *Microcontroladores Uma Abordagem Pratica e Objetiva*. São Paulo, Editora ERICA, 2006.

LUCENA, Gustavo Gomes. Monografia: Automação Residencial por comando de voz utilizando microcontrolador. 2º semestre de 2006.

APÊNDICE A – CODIGO FONTE – FORMULÁRIO PRINCIPAL.FRM

```
'**
' PROJETO FINAL DE CONCLUSÃO DE CURSO
' CONTROLE DO APARELHO DE TELEVISÃO
' VIA COMANDO DE VOZ UTILIZANDO MICROCONTROLADOR
' FORMULÁRIO PRINCIPAL.FRM
' DESENVOLVIDO POR LEONARDO LINS DE ALBUQUERQUE LIMA
'**
Option Explicit

Private Sub btnDesativar_Click()
    'Torna visivel o botão Desativar
    Serial.Abre_Conexao False
    btnDesativar.Visible = False
    'Torna invisivel o botão inicializar
    btnInicializar.Visible = True
    Timer1.Enabled = False
    Timer1_Timer
End Sub

Private Sub btnInicializar_Click()
    'Torna invisivel o botão Inicializar
    btnInicializar.Visible = False
    'Torna visivel o botão Desativar
    btnDesativar.Visible = True
    'Limpa a caixa de comando
    Principal.txtEntra.Text = ""
    'Abre a conexão Serial
    Serial.Abre_Conexao True
    'Inicializa loop de tempo
    Timer1.Enabled = True
End Sub

Private Sub BtnLimparReg_Click()
    'Limpa a variável de historio que alimenta o quadro de registro
    Historico = ""
    'Limpa o quadro de registro
    txtLog.Text = ""
End Sub

Private Sub Form_Load()
    '* Inicio do programa
    'Limpa variável de historico que alimenta o quadro de registro
    Historico = ""
    'Limpa as variaveis de controle
    Limpar_Variaveis
    'Constante de novo paragrafo no quadro de registro
```

```
    Paragrafo = Chr(13) & Chr(10)
    'Torna o botão Desativar invisivel
    btnDesativar.Visible = False
    'Torna o botão Inicializar visivel
    btnInicializar.Visible = True
    'Inicializa controle do loop de tempo
    Timer1.Enabled = False
    'Coloca o loop da controle de tempo em 2.5 segundos
    Timer1.Interval = 2500
    Timer2.Enabled = False
    Timer2.Interval = 8500
    Banco.Abre_Conexao
End Sub

Private Sub Form_Unload(Cancel As Integer)
    'Ao fechar o programa, desativa conexão serial
    Serial.Abre_Conexao False
    Banco.Fecha_Conexao
End Sub

Private Sub Sair_Click()
    'Sai do aplicativo
    End
End Sub

Private Sub Timer1_Timer()
    'Envia o(s) comando(s) para o modulo serial para validação
    Verifica_Comando Principal.txtEntra.Text
    'Atualiza o quadro de registro
    Principal.txtLog.Text = Historico
End Sub

Private Sub Timer2_Timer()
    'Limpa as variáveis de controle
    Limpar_Variaveis
End Sub
```

APÊNDICE B – CODIGO FONTE – MÓDULO BANCO.BAS

```
'**
' PROJETO FINAL DE CONCLUSÃO DE CURSO
' CONTROLE DO APARELHO DE TELEVISÃO
' VIA COMANDO DE VOZ UTILIZANDO MICROCONTROLADOR
' MÓDULO BANCO.BAS
' DESENVOLVIDO POR LEONARDO LINS DE ALBUQUERQUE LIMA
'**
Option Explicit
Private rs As ADODB.Recordset
Private bd As ADODB.Connection
Const strcon = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=D:\Projeto_Final\VB\Banco.mdb;Persist Security Info=False"

Public Sub Insere_Banco(Palavra As String)
   'Se houver algum erro ir para a linha ErroBanco:
   On Error GoTo ErroBanco
   Dim SQL As String
   'Seta a variável bd como nova intancia de conexão
   Set rs = New ADODB.Recordset
   'Define o tipo de conexão
   rs.CursorLocation = adUseClient
   'String SQL para selecionar a Tabela: Tabela
   SQL = "SELECT * FROM TABELA"
   'Abre a gravação pasasndo os parametros de configuração
   rs.Open SQL, strcon, adOpenForwardOnly, adLockOptimistic
   'Adiciona nova linha ao banco de dados
   rs.AddNew
      'Atribui ao campo Comando da tabela a palavra
      rs("Comando") = Palavra
   'Atualiza o banco de dados
   rs.Update
   'Fecha a gravação ativa
   Set rs.ActiveConnection = Nothing
   Exit Sub

ErroBanco:
   'mostra o número do erro e a descrição
   MsgBox Err.Number & " - " & Err.Description
   'Fecha a gravação ativa
   Set rs.ActiveConnection = Nothing
   'Fecha a conexao com o banco de dados
   Set bd = Nothing
End Sub


Public Function Comando_liga(Palavra As String, Tipo As Integer) As Boolean
   Dim SQL As String
   Comando_liga = False
```

```
    SQL = "SELECT Comando FROM Comandos WHERE ID_Tipo = " & Tipo
    'Seta a variável bd como nova intancia de conexão
    Set rs = New ADODB.Recordset
    rs.Open SQL, strcon, adOpenStatic, adLockReadOnly
    Do While Not rs.EOF
        If rs("Comando") = Palavra Then
            Comando_liga = True
            Exit Do
        Else
            rs.MoveNext
        End If
    Loop
End Function

Sub Abre_Conexao()
    Set bd = New ADODB.Connection
    'Abre a conexão passando as especificações
    bd.Open strcon
    'seta a variável rs como nova intancia de Gravação
End Sub

Sub Fecha_Conexao()
    'Fecha a conexao com o banco de dados
    Set bd = Nothing
End Sub
```

APÊNDICE C – CODIGO FONTE – MÓDULO SERIAL.BAS

```
'**
' PROJETO FINAL DE CONCLUSÃO DE CURSO
' CONTROLE DO APARELHO DE TELEVISÃO
' VIA COMANDO DE VOZ UTILIZANDO MICROCONTROLADOR
' MÓDULO SERIAL.BAS
' DESENVOLVIDO POR LEONARDO LINS DE ALBUQUERQUE LIMA
'**
Option Explicit
Global Historico As String
Global Paragrafo As String
Private varON, varCANAL, varSOM, varTV As Boolean
Private Cmd1, Cmd2, Cmd3 As String, TpCmd2, TpCmd3 As Integer

Public Sub Abre_Conexao(n As Boolean)
   On Error GoTo ErrorConexao
   Principal.MSComm1.PortOpen = n
   If n = True Then
      Historico = Historico & "Porta Serial Inicializada:" & _
            Paragrafo & "Baud Rate: 9600, DataBits: 8, Stopbit: 1, Paridade:0" & _
            Paragrafo
   Else
      Historico = Historico & "Porta Serial Desativada" & Paragrafo
   End If
   Exit Sub
ErrorConexao:
   Historico = Historico & "Erro ao conectar à porta serial"
End Sub

Private Sub EnviaTxt(txt As String)
   Principal.MSComm1.Output = txt
End Sub

Public Sub Verifica_Comando(txtCMD As String)
   Dim Palavra As String
   txtCMD = UCase(txtCMD) & " "
   If Principal.MSComm1.PortOpen = True Then
      Do While Len(txtCMD) > 0 And (Left(txtCMD, 1) < "0" Or Left(txtCMD, 1) > "9")
         If InStr(" ", Left(txtCMD, 1)) = 0 Then
            Palavra = Palavra & Left(txtCMD, 1)
         Else
            If Len(Palavra) >= 1 Then
               Palavra = Retira_Acento(Palavra)
               Ordena_Comando Palavra
               Banco.Insere_Banco Palavra
            End If
            Palavra = ""
         End If
```

```
        txtCMD = Mid(txtCMD, 2)
      Loop
      If Cmd1 <> "" Then Executa_Comando Cmd1
      Cmd1 = ""
      If Cmd2 <> "" Then Executa_Comando Cmd2
      Cmd2 = ""
      If Cmd3 <> "" Then Executa_Comando Cmd3
      Cmd3 = ""
   End If
   Principal.txtEntra.Text = ""
   Principal.txtEntra.SetFocus
End Sub

Private Sub Executa_Comando(ByVal Palavra As String)
   If Cmd1 <> "" Then
      varTV = True
      Principal.Timer2.Enabled = True
      Principal.Online.BackColor = &HFF00&
      Beep
   ElseIf Cmd2 <> "" And TpCmd2 = 1 Then
         If varON = False And varTV = True Then
            varON = True
            EnviaTxt 1
            Historico = Historico & "cmd: Liga!" & Paragrafo
         ElseIf varON = True And varTV = True Then
            Historico = Historico & "Falha! Dispositivo já está ligado!" & Paragrafo
         End If
    ElseIf Cmd2 <> "" And TpCmd2 = 2 Then
         If varON = True And varTV = True Then
             varON = False
             EnviaTxt 2
             Historico = Historico & "Cmd: Desliga!" & Paragrafo
          ElseIf varON = False And varTV = True Then
             Historico = Historico & "Falha! Dispositivo está desligado!" & Paragrafo
          End If
   ElseIf Cmd2 <> "" And TpCmd2 = 3 Then
         If varTV = True And varSOM = True Then
            varSOM = False
         End If
         If varTV = True Then
            varCANAL = True
            Principal.Online2.Text = "CANAL"
            Principal.Online2.BackColor = &HFF00&
            Reinicia_Timer2
            Beep
         End If
   ElseIf Cmd2 <> "" And TpCmd2 = 4 Then
         If varTV = True And varCANAL = True Then
            varCANAL = False
         End If
```

```
            If varTV = True Then
               varSOM = True
               Principal.Online2.Text = "SOM"
               Principal.Online2.BackColor = &HFF00&
               Reinicia_Timer2
               Beep
            End If
      ElseIf Cmd3 <> "" And TpCmd3 = 5 Then
            Cmd3 = ""
            If varTV = True And varCANAL = True And varSOM = False Then
               EnviaTxt 3
               Historico = Historico & "cmd:+ Canal!" & Paragrafo
               Reinicia_Timer2
            ElseIf varTV = True And varCANAL = False And varSOM = True Then
               EnviaTxt 5
               Historico = Historico & "cmd:+ SOM!" & Paragrafo
               Reinicia_Timer2
            End If
      ElseIf Cmd3 <> "" And TpCmd3 = 6 Then
            Cmd3 = ""
            If varTV = True And varCANAL = True And varSOM = False Then
               EnviaTxt 4
               Historico = Historico & "cmd:- Canal!" & Paragrafo
               Reinicia_Timer2
            ElseIf varTV = True And varCANAL = False And varSOM = True Then
               EnviaTxt 6
               Historico = Historico & "cmd:- Som!" & Paragrafo
               Reinicia_Timer2
            End If
      End If
End Sub

Private Function Retira_Acento(Palavra As String) As String
      Dim Letra As String, i As Integer
      For i = 1 To Len(Palavra)
         Letra = Mid(Palavra, i, 1)
         If Letra = "Ã" Or Letra = "Á" Or Letra = "Â" Or Letra = "À" Then
            Letra = "A"
         ElseIf Letra = "É" Or Letra = "Ê" Then
            Letra = "E"
         ElseIf Letra = "Í" Or Letra = "Ï" Then
            Letra = "I"
         ElseIf Letra = "Ó" Or Letra = "Ô" Or Letra = "Õ" Then
            Letra = "O"
         ElseIf Letra = "Ú" Or Letra = "Ü" Then
            Letra = "U"
         ElseIf Letra = "Ç" Then
            Letra = "C"
         End If
         Retira_Acento = Retira_Acento & Letra
```

```
    Next
End Function

Sub Reinicia_Timer2()
    Principal.Timer2.Enabled = False
    Principal.Timer2.Enabled = True
End Sub

Public Sub Limpar_Variaveis()
    Principal.Timer2.Enabled = False
    Cmd1 = ""
    Cmd2 = ""
    Cmd3 = ""
    varTV = False
    varCANAL = False
    varSOM = False
    Principal.Online.BackColor = &HFF&
    Principal.Online2.BackColor = &HFF&
    Principal.Online2.Text = ""
End Sub

Private Sub Ordena_Comando(Palavra As String)
    If (Banco.Comando_liga(Palavra, 8) = True) Then
        Cmd1 = Palavra
    End If
    If (Banco.Comando_liga(Palavra, 1) = True) Then
        Cmd2 = Palavra
        TpCmd2 = 1
    End If
    If (Banco.Comando_liga(Palavra, 2) = True) Then
        Cmd2 = Palavra
        TpCmd2 = 2
    End If
    If (Banco.Comando_liga(Palavra, 3) = True) Then
        Cmd2 = Palavra
        TpCmd2 = 3
    End If
    If (Banco.Comando_liga(Palavra, 4) = True) Then
        Cmd2 = Palavra
        TpCmd2 = 4
    End If
    If (Banco.Comando_liga(Palavra, 5) = True) Then
        Cmd3 = Palavra
        TpCmd3 = 5
    End If
    If (Banco.Comando_liga(Palavra, 6) = True) Then
        Cmd3 = Palavra
        TpCmd3 = 6
    End If
End Sub
```

APÊNDICE D – CÓDIGO FONTE – MICROCONTROLADOR.ASM

```
;**
; PROJETO FINAL DE CONCLUSÃO DE CURSO
; CONTROLE DO APARELHO DE TELEVISÃO
; VIA COMANDO DE VOZ UTILIZANDO MICROCONTROLADOR
; MICRCONTROLADOR.ASM
; DESENVOLVIDO POR LEONARDO LINS DE ALBUQUERQUE LIMA
;**

#INCLUDE P16F877A.INC

__CONFIG _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF & _BODEN_ON & _LVP_OFF

#DEFINE BANK0 BCF STATUS,RP0      ; Declarando comando para mudar de banco do microcontrolador PIC
#DEFINE BANK1 BSF STATUS,RP0      ; Declarando comando para mudar de banco do microcontrolador PIC

#DEFINE ENB       PORTB,5
#DEFINE RS  PORTB,4

CBLOCK      20H
            SALVA_W    ; Salva conteúdo de W nas interrupções
            SALVA_S    ; Salva conteúdo de STATUS nas interrupções
            TEMPO1     ; Usado nas rotinas de tempo
            TEMPO2     ; Usado nas rotinas de tempo
            CHAR ; Caracter ou comando a ser enviado p/ LCD
            DADO; Caracter recebido via RS-232
            TIMEOUTDB ; Caracter de validacao
ENDC

ORG   000H
            GOTO INICIO

; ***************** Tratamento das interrupções ******************

ORG   004H
            MOVWF      SALVA_W          ; Salva contexto
            SWAPF      STATUS,W
            MOVWF      SALVA_S

            MOVFRCREG,W          ; Lê dado recebido e zera flags
            MOVWF      DADO

FIMINT:
            SWAPF      SALVA_S,W
            MOVWF      STATUS
            SWAPF      SALVA_W,F
```

```
                SWAPF       SALVA_W,W
RETFIE

; ******************** Inicialização do sistema ********************

INICIO:
                BANK1
                MOVLW       B'00001110'
                MOVWF       TRISA
                MOVLW       0C0H
                MOVWF       TRISB
                MOVLW       0BFH            ; RC7 = RX, RC6 = TX,  1011 1111
                MOVWF       TRISC
                MOVLW       B'00000111'     ; Porta A como sinais digitais
                MOVWF       ADCON1
                MOVLW       24
                MOVWF       TXSTA                   ; Modo assíncrono, 8 bits, TX habilitada
                MOVLW       .25
                MOVWF       SPBRG                   ; Baud rate = 9600 bps
                MOVLW       B'11000000'
                MOVWF       INTCON                  ; Habilita int. periféricos
                MOVLW       20
                MOVWF       PIE1            ; Habilita int. RX serial
                CLRF  PIE2
                BANK0

                MOVLW       90
                MOVWF       RCSTA                   ; Serial habilitada, recepcao continua, 8
bits
                MOVFRCREG,W             ; Limpa registrador e flags de recepcao

; ************ Inicializacao do display ************

                CALL LP_250MS

        MOVLW   28          ; Interface 4 bits, 2 linhas, caracter 5x7
        CALL SEND_CMD

        MOVLW       0C              ; Cursor desligado
        CALL SEND_CMD

        MOVLW       06              ; Deslocamento para a direita
        CALL SEND_CMD


LACO_INICIAL
        MOVLW       01              ; Apaga display
        CALL SEND_CMD
        MOVLW       0x80  ; Endereca a DDRAM do LCD para linha 1 e coluna 1 do
display
```

```
                CALL SEND_CMD

                MOVLW       'A'             ; Escreve "RX: " no LCD
                CALL SEND_CHAR
                MOVLW       'G'             ; Escreve "RX: " no LCD
                CALL SEND_CHAR
                MOVLW       'U'             ; Escreve "RX: " no LCD
                CALL SEND_CHAR
                MOVLW       'A'             ; Escreve "RX: " no LCD
                CALL SEND_CHAR
                MOVLW       'R'
                CALL SEND_CHAR
                MOVLW       'D'             ; Escreve "RX: " no LCD
                CALL SEND_CHAR
                MOVLW       'A'
                CALL SEND_CHAR
                MOVLW       'N'
                CALL SEND_CHAR
                MOVLW       'D'
                CALL SEND_CHAR
                MOVLW       'O'
                CALL SEND_CHAR
                MOVLW       ' '
                CALL SEND_CHAR
                MOVLW       'C'
                CALL SEND_CHAR
                MOVLW       'M'
                CALL SEND_CHAR
                MOVLW       'D'
                CALL SEND_CHAR
                MOVLW       '.'
                CALL SEND_CHAR
                MOVLW       '.'
                CALL SEND_CHAR

;*************Validação*******************
        CLRF DADO

RECEBE_SERIAL
                MOVLW   '1'                 ; coloca o valor de 1 na variavel W
                SUBWF   DADO,W      ; subtrai variavel DADO do Valor da variavel W work
                BTFSC   STATUS,Z  ; verifica se o status Z esta clear: se sim vai (goto) funcao
VALIDACAO_OK
                GOTO            VALIDACAO_LIGA

                MOVLW   '2'                 ; coloca o valor de 2 na variavel W
                SUBWF   DADO,W      ; subtrai variavel DADO do Valor da variavel W work
                BTFSC   STATUS,Z  ;
                GOTO            VALIDACAO_DESLIGA
```

```
                MOVLW   '3'                     ; coloca o valor de 3 na variavel W
                SUBWF   DADO,W      ; subtrai variavel DADO do Valor da variavel W work
                BTFSC   STATUS,Z ;
                GOTO            VALIDACAO_SOBE_CANAL

                MOVLW   '4'                     ; coloca o valor de 3 na variavel W
                SUBWF   DADO,W      ; subtrai variavel DADO do Valor da variavel W work
                BTFSC   STATUS,Z ;
                GOTO            VALIDACAO_DESCE_CANAL

                MOVLW   '5'                     ; coloca o valor de 3 na variavel W
                SUBWF   DADO,W      ; subtrai variavel DADO do Valor da variavel W work
                BTFSC   STATUS,Z ;
                GOTO            VALIDACAO_MAIS_SOM

                MOVLW   '6'                     ; coloca o valor de 3 na variavel W
                SUBWF   DADO,W      ; subtrai variavel DADO do Valor da variavel W work
                BTFSC   STATUS,Z ;
                GOTO            VALIDACAO_MENOS_SOM

                MOVLW   d'5'                            ; coloca o valor de 5 na variavel W
                SUBWF   TIMEOUTDB,W             ; subtrai variavel TIMEOUTDB do Valor
da variavel W work
                BTFSC   STATUS,Z            ; verifica se o status Z esta clear: se sim vai (goto)
funcao TENTE_DE_NOVO
                GOTO            RECEBE_SERIAL
                INCF    TIMEOUTDB,F             ;       incrementa      o       valor       da
variavel TIMEOUTDB

                GOTO            LACO_INICIAL


; ***************** Laço principal *****************

VALIDACAO_LIGA
                MOVLW       01                              ; coloca o valor 1 na variavel W
para ser enviado a funcao SEND_CMD para Apagar display
                CALL SEND_CMD
                MOVLW       0x80                    ; Endereca a DDRAM do LCD para linha
1 e coluna 1 do display
                CALL SEND_CMD

                MOVLW       'L'             ; Escreve "LIGA: " no LCD
                CALL SEND_CHAR
                MOVLW       'I'
                CALL SEND_CHAR
                MOVLW       'G'
                CALL SEND_CHAR
                MOVLW       'A'
                CALL SEND_CHAR
```

```
        MOVLW   '!'
        CALL SEND_CHAR
        NOP
        MOVLW   'A'
        MOVWF   TXREG           ; Transmite caracter "A"
        CALL ESPERA_1SEG
        CALL LP_250MS
        CALL LP_250MS
        GOTO    LACO_INICIAL

VALIDACAO_DESLIGA
        MOVLW   01              ; coloca o valor 1 na variavel W
para ser enviado a funcao SEND_CMD para Apagar display
        CALL SEND_CMD
        MOVLW   0x80            ; Endereca a DDRAM do LCD para linha
1 e coluna 1 do display
        CALL SEND_CMD

        MOVLW   'D'         ; Escreve "DESLIGA: " no LCD
        CALL SEND_CHAR
        MOVLW   'E'
        CALL SEND_CHAR
        MOVLW   'S'
        CALL SEND_CHAR
        MOVLW   'L'
        CALL SEND_CHAR
        MOVLW   'I'
        CALL SEND_CHAR
        MOVLW   'G'
        CALL SEND_CHAR
        MOVLW   'A'
        CALL SEND_CHAR
        MOVLW   '!'
        CALL SEND_CHAR
        NOP
        MOVLW   'A'
        MOVWF   TXREG           ; Transmite caracter "A"
        CALL ESPERA_1SEG
        CALL LP_250MS
        CALL LP_250MS
        GOTO    LACO_INICIAL

VALIDACAO_SOBE_CANAL
        MOVLW   01              ; coloca o valor 1 na variavel W
para ser enviado a funcao SEND_CMD para Apagar display
        CALL SEND_CMD
        MOVLW   0x80            ; Endereca a DDRAM do LCD para linha
1 e coluna 1 do display
        CALL SEND_CMD
```

```
        MOVLW   '+'             ; Escreve "+CANAL: " no LCD
        CALL SEND_CHAR
        MOVLW   ' '
        CALL SEND_CHAR
        MOVLW   'C'
        CALL SEND_CHAR
        MOVLW   'A'
        CALL SEND_CHAR
        MOVLW   'N'
        CALL SEND_CHAR
        MOVLW   'A'
        CALL SEND_CHAR
        MOVLW   'L'
        CALL SEND_CHAR
        MOVLW   '!'
        CALL SEND_CHAR
        NOP
        MOVLW   'A'
        MOVWF   TXREG               ; Transmite caracter "A"
        CALL ESPERA_1SEG
        CALL LP_250MS
        CALL LP_250MS
        GOTO    LACO_INICIAL

VALIDACAO_DESCE_CANAL
        MOVLW   01                  ; coloca o valor 1 na variavel W
para ser enviado a funcao SEND_CMD para Apagar display
        CALL SEND_CMD
        MOVLW   0x80                ; Endereca a DDRAM do LCD para linha
1 e coluna 1 do display
        CALL SEND_CMD

        MOVLW   '-'             ; Escreve "-CANAL: " no LCD
        CALL SEND_CHAR
        MOVLW   ' '
        CALL SEND_CHAR
        MOVLW   'C'
        CALL SEND_CHAR
        MOVLW   'A'
        CALL SEND_CHAR
        MOVLW   'N'
        CALL SEND_CHAR
        MOVLW   'A'
        CALL SEND_CHAR
        MOVLW   'L'
        CALL SEND_CHAR
        MOVLW   '!'
        CALL SEND_CHAR
        NOP
        MOVLW   'A'
```

```
        MOVWF   TXREG           ; Transmite caracter "A"
        CALL ESPERA_1SEG
        CALL LP_250MS
        CALL LP_250MS
        GOTO    LACO_INICIAL

VALIDACAO_MAIS_SOM
        MOVLW   01              ; coloca o valor 1 na variavel W
para ser enviado a funcao SEND_CMD para Apagar display
        CALL SEND_CMD
        MOVLW   0x80            ; Endereca a DDRAM do LCD para linha
1 e coluna 1 do display
        CALL SEND_CMD

        MOVLW   '+'         ; Escreve "+ SOM: " no LCD
        CALL SEND_CHAR
        MOVLW   ' '
        CALL SEND_CHAR
        MOVLW   'S'
        CALL SEND_CHAR
        MOVLW   'O'
        CALL SEND_CHAR
        MOVLW   'M'
        CALL SEND_CHAR
        MOVLW   '!'
        CALL SEND_CHAR
        NOP
        MOVLW   'A'
        MOVWF   TXREG           ; Transmite caracter "A"
        CALL ESPERA_1SEG
        CALL LP_250MS
        CALL LP_250MS
        GOTO    LACO_INICIAL

VALIDACAO_MENOS_SOM
        MOVLW   01              ; coloca o valor 1 na variavel W
para ser enviado a funcao SEND_CMD para Apagar display
        CALL SEND_CMD
        MOVLW   0x80            ; Endereca a DDRAM do LCD para linha
1 e coluna 1 do display
        CALL SEND_CMD

        MOVLW   '-'         ; Escreve "- SOM: " no LCD
        CALL SEND_CHAR
        MOVLW   ' '
        CALL SEND_CHAR
        MOVLW   'S'
        CALL SEND_CHAR
        MOVLW   'O'
        CALL SEND_CHAR
```

```
                MOVLW   'M'
                CALL SEND_CHAR
                MOVLW   '!'
                CALL SEND_CHAR
                NOP
                MOVLW   'A'
                MOVWF   TXREG               ; Transmite caracter "A"
                CALL ESPERA_1SEG
                CALL LP_250MS
                CALL LP_250MS
                GOTO    LACO_INICIAL

; ***************** Envia comandos e dados p/ LCD *****************

; -> Envia comando
SEND_CMD:
                MOVWF   CHAR        ; Salva comando
        BCF     RS          ; Coloca o LCD em modo comando
                GOTO ENVIA

; -> Envia dado
SEND_CHAR:
                MOVWF   CHAR        ; Salva caracter
      BSF     RS            ; Coloca o LCD em modo dados
ENVIA:          MOVLW   0F0
                ANDWF   PORTB,F
                SWAPF   CHAR,W
                ANDLW   0F          ; Nibble mais significativo
                IORWF   PORTB,F

            BSF ENB         ; Gera sinal para o LCD
                NOP
                NOP
                BCF    ENB

                MOVLW   0F0
                ANDWF   PORTB,F
                MOVF CHAR,W
                ANDLW   0F          ; Nibble menos significativo
                IORWF   PORTB,F

            BSF ENB         ; Gera sinal para o LCD
                NOP
                NOP
                BCF    ENB
                GOTO LP_2MS

; ************************ Laços de tempo ************************
ESPERA_1SEG
                CALL LP_250MS
```

```
                CALL LP_250MS
                CALL LP_250MS
                CALL LP_250MS
RETURN

LP_250MS:
                MOVLW       .250
                MOVWF       TEMPO2
                GOTO LP_1MS
LP_2MS:
                MOVLW       .2
                MOVWF       TEMPO2
LP_1MS:
                MOVLW       .250
                MOVWF       TEMPO1
LOOP:
                NOP
                DECFSZ      TEMPO1,F
                GOTO LOOP
                DECFSZ      TEMPO2,F
                GOTO LP_1MS
RETURN
END
```