

Sergio Ricardo Ferrazoli

Interface Braille Automatizada

Londrina

2006

Sergio Ricardo Ferrazoli

Interface Braille Automatizada

Dissertação apresentada ao Curso de Pós-graduação, em Engenharia Elétrica, da Universidade Estadual de Londrina, como requisito parcial à obtenção do título de Mestre.

Orientador:

Prof. Dr. Walter Germanovix

UNIVERSIDADE ESTADUAL DE LONDRINA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Londrina

2006

Dissertação de Mestrado sob o título *Interface Braille Automatizada*, apresentada por Sergio Ricardo Ferrazoli e aprovada em 18 de dezembro de 2006, em Londrina, Estado do Paraná, pela banca examinadora constituída pelos doutores:

Prof. Dr. Walter Germanovix
Orientador - UEL

Prof. Dr. Ernesto F. Ferreyra Ramirez
UEL

Prof. Dr. Aparecido Augusto de Carvalho
UNESP

Aos meus filhos Daniel e Gabriel, aos quais desejo que a vontade de aprender e o amor ao próximo sejam a fonte de seus desenvolvimentos.

Agradecimentos

Agradeço a Deus por me permitir chegar até aqui, e ter colocado em minha vida muitas pessoas maravilhosas, como anjos, dentro deste novo caminho que Ele me proporcionou trilhar.

Agradeço aos meus filhos Daniel e Gabriel, e minha esposa Aranele, pelo amor, e compreensão às minhas faltas como pai e marido durante minha caminhada na busca da conquista deste sonho.

Agradeço ao meu pai por toda dedicação, amor e apoio, sem os quais jamais chegaria até aqui.

Ao Prof. Dr. Walter Germanovix pela sua competência e pelo seu comprometimento com a realização deste trabalho; por ser um exemplo de humanidade e profissionalismo com inspiradora paixão pelo trabalho.

Não poderia deixar de citar os amigos Marcos Gilmar Zahailo, James Arnold, Sammy Wladsem Faria do Nascimento e Mariana Peres de Oliveira que, dispensaram muita dedicação em me auxiliar nos momentos que, me encontrava em pleno deserto de idéias.

Aos colegas de pós-graduação, pelo companheirismo e amizade, que marcaram profundamente a memória; nas jornadas de estudo, nas conversas descontraídas e nas palavras de incentivo.

Aos Deficientes Visuais que tive a oportunidade de conhecer e conviver, principalmente ao meu novo amigo Lucildo Teodoro.

A todos do CAEDV - Centro de Atendimento Especializado para Deficientes Visuais, Escola Municipal Dr. Borges de Medeiros Educação Infantil e Ensino Fundamental, à Diretora Solange Fagotti Pagliarini e às professoras Rute Muniz Silveira, Maricélia Travessini, Naziria Maria Teixeira e Regina Meris Vitkoski, pela alegria, apoio e incentivo.

À minha prima Maria Isabel, que muito se empenhou em auxiliar nas revisões, fornecendo sugestões para tornar o trabalho melhor.

A todos aqueles que, por um lapso, esqueci de mencionar e que foram importantes para a realização deste trabalho.

Sumário

Glossário	p. viii
Lista de Abreviaturas	p. ix
Lista de Figuras	p. x
Lista de Tabelas	p. xiii
Resumo	p. xiv
Abstract	p. xv
1 Introdução	p. 1
1.1 Objetivos	p. 2
1.1.1 Objetivo Específico	p. 2
1.1.2 Objetivos Gerais	p. 2
1.1.3 Resultados Esperados	p. 3
1.2 Procedimentos metodológicos - métodos e técnicas	p. 3
1.2.1 Software	p. 3
1.2.2 Hardware	p. 4
1.3 Organização da dissertação	p. 4
1.4 Resumo do Capítulo 1	p. 4
2 A solução proposta	p. 6
2.1 O código Braille	p. 6
2.1.1 Histórico do Braille	p. 6
2.1.2 Funcionamento do Código Braille	p. 8
2.2 Interface Braille Automatizada	p. 11

2.2.1	Aplicações do Sistema Proposto	p. 12
2.2.2	Vantagens do Sistema Proposto	p. 14
2.2.3	Desvantagens	p. 14
2.3	Resumo do Capítulo 2	p. 15
3	Descrição da Interface Braille Automatizada	p. 18
3.1	Software	p. 20
3.1.1	Desenvolvimento do Software SoftBraille	p. 20
3.1.1.1	Rotina da porta serial (MSComm)	p. 20
3.1.1.2	Rotina que executa os sons (PlaySound)	p. 20
3.1.1.3	Alfabetiza	p. 21
3.1.1.4	Capturando a tecla pressionada (KeyDown)	p. 21
3.1.1.5	Conversão de arquivos	p. 21
3.2	Hardware	p. 22
3.2.1	Desenvolvimento do Hardware	p. 22
3.2.2	A escolha do Microcontrolador	p. 23
3.2.3	O Software Embarcado	p. 24
3.2.4	Funcionamento do Hardware	p. 24
3.2.4.1	A Fonte de Alimentação	p. 24
3.2.4.2	A Comunicação Serial	p. 25
3.2.4.3	Módulo de controle	p. 26
3.2.4.4	O teclado do Hardware	p. 29
3.2.4.5	O circuito de acionamento	p. 29
3.2.4.6	Circuitos Adicionais	p. 32
3.2.4.7	A construção do protótipo eletro-mecânico da celinha	p. 32
3.3	Resumo do capítulo 3	p. 34
4	Validação	p. 39
4.1	Teste da Interface Braille Automatizada por um DV	p. 39
4.1.1	O início	p. 40
4.1.2	A sensação tátil	p. 40

4.1.3	Comparação entre IBA e impressão Braille em papel pelo DV	p. 40
4.1.4	O manuseio do SoftBraille pelo DV	p. 40
4.1.5	Sugestões durante a utilização	p. 41
4.2	Conclusão da validação	p. 41
4.3	Resumo do Capítulo 4	p. 41
5	Estado atual e trabalhos futuros	p. 42
5.1	Inovação tecnológica	p. 42
5.1.1	Depósito de Patente	p. 42
5.2	Perspectivas de mercado	p. 42
5.3	Futuras contribuições ao trabalho	p. 43
5.4	Resumo do Capítulo 5	p. 43
6	Conclusão	p. 44
	Referências Bibliográficas	p. 45
	Apêndice A – Rotinas principais do Visual Basic	p. 49
A.1	Formulário principal (FrmPrincipal)	p. 49
A.2	Formulário de mensagens (Frm_Msg_Box)	p. 59
A.3	Módulo Converte	p. 61
A.4	Módulo de declarações de variáveis (Declarar)	p. 63
A.5	Módulo que executa os sons (FunçãoPlaySound)	p. 64
	Apêndice B – Rotinas principais do Software Embarcado	p. 65
B.1	Declarações	p. 65
B.2	Rotina Principal	p. 68
B.3	Conversão Braille	p. 71
B.4	Outras Rotinas	p. 77
B.5	Comunicação Serial	p. 83
	Apêndice C – Manual do SoftBraille	p. 85
C.1	Área visual do programa	p. 85

C.1.1	Os botões, sua teclas de atalho e suas funções detalhadas	p. 85
C.1.2	As caixas de texto e suas funções	p. 87
C.2	Funcionamento do software	p. 88
C.2.1	Opção Converte	p. 90
Anexo A	- Poster apresentado na Semana de Ciência e Tecnologia da UEL	p. 91
Anexo B	- Declaração de participação no I Seminário de Inserção Social - Educação e Trabalho das Pessoas com Deficiência Visual	p. 93
Anexo C	- Cópia do Depósito de Patente	p. 95
Anexo D	- Perguntas feitas a usuários Deficientes Visuais e Videntes durante o I Seminário de Inserção Social - Educação e Trabalho das Pessoas com Deficiência Visual.	p. 99
Anexo E	- Resumo impresso em Braille	p. 103

Glossário

ASCII: do inglês (American Standard Code for Information Interchange) é um conjunto de códigos para o computador representar números, letras, pontuação e outros caracteres

Bobina: componente eletromecânico destinado a criar uma campo magnético quando energizado.

Buzzer: componente utilizado para emitir um som

Celinha: nome que se dá a um conjunto de seis pontos que representam um caractere no alfabeto Braille

DEMO: termo utilizado em softwares para ilustrar a utilização com tempo limitado para demonstração

Form: tela onde são introduzidos os componentes para programação visual

Fosfenes: uma luz ou uma sensação de luz que aparece ocasionalmente no campo de visão, ao fechar ou abrir os olhos

Ground: terra. No circuito representa a alimentação negativa da fonte

Kbps: um kilobit por segundo (kbps or kbit/s) é uma unidade de transmissão de dados igual a 1.000 bits por segundo

KeyDown: código do Microsoft Visual Basic que armazena um certo dado ao se pressionar uma determinada tecla

Label: nome de um componente do Microsoft Visual Basic que significa rótulo

LED: termo do inglês Light Emission Diode. Pequeno componente tipo diodo emissor de luz

MSComm: nome de um componente do Microsoft Visual Basic que controla a porta serial de um microcomputador

PlaySound: rotina criada no SoftBraille para executar sons

Reglete: régua utilizada pelo deficiente visual para escrever em Braille

Solenóide: o mesmo que bobina

Vidente: pessoa de visão normal. Termo utilizado pelo Deficiente Visual para se referir aos que vêem

Lista de Abreviaturas

ASCII - American Standard Code for Information Interchange.

CAEDV - Centro de Atendimento Especializado para Deficientes Visuais.

CCT - Circuito.

CI - Circuito Integrado.

DV - Deficiente Visual.

GND - Ground.

IBA - Interface Braille Automatizada.

Kbps - Kilo bits por segundo.

LED - Light Emission Diode.

NU - Não utilizado.

Lista de Figuras

2.1	Mãos tocando o papel e através do tato proporcionando a leitura. . . .	p. 6
2.2	Louis Braille.	p. 7
2.3	Código Braille, para representação do alfabeto. Cada conjunto de círculos maiores, que são em alto relevo, representa um caractere.	p. 7
2.4	Reglete - Régua utilizada para a escrita Braille.	p. 8
2.5	Nas celinhas os pontos são numerados de 1 a 6 e a combinação deles resulta no Código Braille.	p. 8
2.6	A combinação dos pontos 1, 2, 4 e 5 formam o grupo de letras de “a” a “j”	p. 9
2.7	Acrescentando-se o ponto 3 ao grupo de letras de “a” a “j” é formado o grupo das letras de “k” a “t”.	p. 9
2.8	Acrescentando-se o ponto 6 ao grupo de letras de “k” a “t” é formado o grupo das letras de “u” a “ú”.	p. 10
2.9	Retira-se o ponto 3 do terceiro grupo das letras de “u” a “ú” e forma-se o grupo das letras de “â” a “w”.	p. 10
2.10	A combinação dos pontos 2, 3, 4, 5 e 6 formam o grupo de caracteres de “ı” a “?”	p. 10
2.11	Os números são formados pela introdução de um sinal especial e as letras de “a” a “j”.	p. 11
2.12	Grupo de sinais.	p. 12
2.13	Celinha e as bobinas que acionarão os pontos do Alfabeto Braille. . . .	p. 13
2.14	Celinha e as bobinas acionadas nos pontos do Alfabeto Braille que representam o caractere “c”.	p. 14
2.15	O DV poderá ler documentos gerados por processadores de texto ou páginas de internet.	p. 15
2.16	Interface Braille Automatizada usada como meio de comunicação para informações relevantes.	p. 15
2.17	Interface Braille Automatizada ligada a um caixa eletrônico.	p. 16

2.18	Interface Braille Automatizada ligada a um microcomputador destinada a alfabetização de um DV.	p. 16
2.19	Outros exemplos de Interface Braille Automatizada ligada independente do microcomputador.	p. 17
3.1	Diagrama em blocos do conceito geral de funcionamento da Interface Braille Automatizada.	p. 18
3.2	Expansão em blocos de todo o sistema.	p. 18
3.3	Diagrama em blocos da interação entre o SoftBraille (presente no sistema computacional) e a Interface Braille Automatizada.	p. 19
3.4	Primeiro circuito montado em <i>protoboard</i> com oito conjuntos de seis LEDs representando oito celinhas.	p. 23
3.5	Desenho das trilhas da placa experimental de controle da Interface Braille Automatizada.	p. 24
3.6	Desenho das trilhas da placa experimental de expansão de celinhas da Interface Braille Automatizada.	p. 25
3.7	Placa experimental de controle da Interface Braille Automatizada montada com os componentes.	p. 26
3.8	Placa experimental de expansão de celinhas da Interface Braille Automatizada montada com os componentes.	p. 27
3.9	Esquema elétrico da fonte de alimentação.	p. 28
3.10	Terminais do CI serial RS 232.	p. 28
3.11	Esquema elétrico do circuito serial.	p. 30
3.12	Esquema elétrico do módulo de controle da Interface Braille Automatizada.	p. 31
3.13	Esquema elétrico do teclado.	p. 32
3.14	Terminais do CI ULN 2004.	p. 32
3.15	Circuito do CI ULN 2004.	p. 33
3.16	Esquema elétrico do circuito de acionamento das bobinas de uma celinha.	p. 34
3.17	Terminais do CI 74HC373.	p. 35
3.18	Circuito do <i>buzzer</i> e LED de verificação do programa.	p. 35
3.19	Conjunto mecânico da Interface Braille Automatizada.	p. 35
3.20	Vista traseira da IBA.	p. 37
3.21	Vista das cabeças, agulhas e molas de guia.	p. 38
4.1	Aluno DV utilizando a Interface Braille Automatizada.	p. 39

C.1	Tela principal do programa SoftBraille.	p. 85
C.2	Tela auxiliar do botão converte.	p. 86
C.3	Complemento da tela auxiliar do botão converte na opção renomear. . .	p. 86
C.4	Exemplo de um arquivo original com formatações e figura.	p. 88
C.5	Exemplo de arquivo convertido em formato texto pronto a ser enviado para a interface.	p. 90

Lista de Tabelas

2.1	Tabela de conjunto de caracteres e pontos	p. 9
3.1	Descrição dos terminais do CI serial RS 232	p. 29
3.2	Descrição dos terminais do microcontrolador 16F877	p. 36
3.3	Descrição dos terminais do CI ULN 2004	p. 37
3.4	Descrição dos terminais do CI 74HC373	p. 37

Resumo

Os Deficientes Visuais, por muito tempo na história, não puderam contar com a leitura e a escrita devido a sua limitação física. Para que esses indivíduos não ficassem privados dessa grande invenção da humanidade, Louis Braille, também privado do sentido da visão, elaborou o código Braille como ferramenta para que os Deficientes Visuais pudessem executar a leitura por meio do sentido do tato, e não da visão. Esse processo de leitura é feito por meio de símbolos impressos em relevo, normalmente em papel, que representam as letras do alfabeto, números e diversos símbolos. Cada símbolo é chamado, em Braille, de celinha. O propósito desse trabalho é desenvolver um protótipo de sistema composto de *hardware* e *software* que possa automatizar duas, ou mais, celinhas do Alfabeto Braille. O *software* do sistema captura uma tecla digitada pelo usuário e envia este caractere por meio de comunicação serial para o *hardware* do sistema que, por sua vez, converte o dado recebido para Braille e o reproduz numa superfície plana para que possa ser lido por meio do tato. Algumas possibilidades de utilização do sistema foram sugeridas neste trabalho, com destaque para o modo alfabetiza. Para avaliação do sistema, foram executados testes práticos no modo alfabetiza com Deficientes Visuais e professores da área de educação especial. Neste modo de operação, a cada caractere digitado pelo usuário, o computador executa o som correspondente e o reproduz como símbolo na Interface Braille Automatizada. O resultado alcançado por essa solução superou as expectativas e demonstrou sua viabilidade técnica e comercial.

Abstract

The Visually Impaired, for much of , has not been able to count on reading and writing due to physical limitations. So that these individuals were not deprived of this great invention of humanity, Louis Braille, also deprived of vision, elaborated the Braille Code as a tool so that the Visually Impaired could read by means of touch, and not sight. This process of reading is made by means of symbols printed matters in relief, normally on paper, that represent the letters of the alphabet, numbers and diverse symbols. Each symbol is called a Braille cell. The intention of this work is to develop a prototype of a system composed of hardware and software that can automate two, or more cell of the Braille alphabet. The software of the system captures a keystroke typed by the user and sends this character by means of serial communication to the hardware of the system that, in turn, converts the data received into Braille and it reproduces it in a plain surface so that it can be read by means of touch. Some possibilities of use of the system have been suggested in this work, with prominence learning the Braille alphabet. For evaluation of the system, learning with visually impaired and teachers in the area of special education has been executed. In operation each character typed by the user, the computer executes the corresponding sound and it reproduces it as symbol in the Automatized Braille Interface. The result achieved for this solution surpassed the expectations and demonstrated its technique and advertising viability.

1 *Introdução*

Existem no Brasil, em um levantamento feito no ano de 1.989 pela UNESCO e pela Organização Mundial de Saúde, aproximadamente 750.000 pessoas, incluindo todas as faixas etárias, com deficiência visual^[2]. Esta deficiência pode ser congênita, adquirida ou perda gradual e ainda pode ser dividida em dois grupos: os cegos que somente podem ler com auxílio do código braille e os de visão sub normal, que conseguem identificar algo como um vulto^[??].

Vários métodos de auxílio ao DV têm sido propostos e implementados. Alguns simples como uma bengala para auxílio a caminhar, com objetivo de vencer os obstáculos, e outros altamente complexos como próteses implantadas para estimulação dos nervos visuais, para geração de fosfenos^[?]. Muito tem sido estudado a fim de encontrar novas ferramentas de auxílio que facilitem a vida de DVs.

Para a leitura de caracteres o auxílio que o DV tem empregado é o código Braille impresso em “alto relevo” que é composto pela combinação de pontos distribuídos em três linhas e duas colunas, com sessenta e três combinações que podem ser números, letras, sinais matemáticos ou símbolos de notação musical. Novas notações para símbolos científicos são constantemente criados. Hoje contamos com uma simbologia específica para a Matemática, Música, Química, Física além de códigos abreviados para taquigrafia ou para anotações de aulas. O método de leitura é do tipo “sentido pelo tato” numa sinalização tátil gravada em superfície de “alto relevo” onde o DV passa suavemente o dedo, normalmente o indicador, para perceber o código daquele caracter^[?].

O computador é uma importante ferramenta para a inclusão do DV na sociedade^[?]. Atualmente a interface com os computadores para o DV é feita com sons utilizando-se sintetizadores de voz e impressoras que imprimem em Braille, cuja vantagem é a facilidade de utilização, pois o microcomputador realmente lê para o usuário DV e as desvantagens de não poder ser utilizado numa sala de aula comum, pois o sistema atrapalharia os demais colegas na mesma sala, sem contar a falta de estimulação para a leitura em Braille. Quanto a impressora, o seu custo é alto e somente são encontradas em algumas associações para DVs. Um exemplo desse modo de utilização do microcomputador pelo DV é o *software* DOSVOX criado pelo núcleo de computação eletrônica da Universidade Federal do Rio de Janeiro(UFRJ)^[?].

Outras interfaces, operando de modo auditivo, auxiliam o DV a atravessar ruas através de sinal auditivo característico em semáforos que indicam que o sinal está

verde ou vermelho para o pedestre. Têm-se desenvolvido outras inúmeras interfaces ou soluções computacionais de auxílio ao DV que são exemplos: Projeto CAP do MEC (Ministério da Educação e Cultura) que busca, através de um sistema computadorizado de interconexão por internet, criar uma infraestrutura nacional para apoio ao DV ^[?] e sistemas de *scanner* que são associados a programas de reconhecimento de texto OCR (*Optical Character Recognition*) a fim de serem utilizados por outros programas que irão ler para o DV ^[?].

Com esta caracterização o Laboratório de Engenharia Biomédica do Departamento de Engenharia Elétrica da UEL, coordenado pelo Prof. Dr. Walter Germanovix, propõe desenvolver uma interface inovadora que permitirá ao DV interpretar os textos gerados por computadores.

A inovação é uma característica marcante da proposta do trabalho, pois gerou inclusive um depósito de patente.

1.1 **Objetivos**

1.1.1 **Objetivo Específico**

Transformar os caracteres de um texto de computador em sinais Braille em uma interface, denominada de Interface Braille Automatizada. Esta interface poderá, ou não, funcionar em tempo real, ou seja, conectada a um microcomputador onde o DV poderá ler o texto que estiver na tela em Braille, ou desconectado do microcomputador, repetindo um determinado texto que tenha sido armazenado na interface. Os textos poderão ser gerados por muitas fontes, sejam elas processadores de textos diversos e arquivos que poderão ser arquivos de livros ou similares.

1.1.2 **Objetivos Gerais**

Os objetivos gerais deste trabalho é apontar possíveis aplicações para o protótipo. Dentre essas aplicações podemos citar:

- Auxiliar a Alfabetização de DVs, com o auxílio do microcomputador e da interface, um dos principais propósitos deste trabalho.
- Implementação da interface em semáforos das cidades, dos quais, através do tato, e de sinais gerados pela interface, o DV poderá receber a mensagem das condições de passagens, além de outras informações.
- Caixas eletrônicos de Bancos poderão ser equipados com a interface para auxiliar o DV em operações bancárias.

- Prédios públicos poderão fornecer informações relevantes aos DVs com a implementação da interface.
- Possibilidade de interpretação de textos gerados via Internet, incluindo a possibilidade de comunicação nos “chats”.

1.1.3 Resultados Esperados

Melhoria da qualidade de vida dos DVs, facilitando sua inclusão social. Segundo pesquisas fornecidas pelo Governo Federal ^[?] a leitura táctil é três vezes mais fatigante que a leitura visual e o material Braille não é tão atraente ao tato como os livros coloridos são para a visão; por isso, não é tão fácil despertar o interesse do leitor em Braille. Portanto este projeto visa melhorar a utilização e tornar o Braille mais atraente para o DV, visto que é um sistema computacional amigável e um sistema automático, diferente de uma simples folha de papel em relevo.

As diretrizes atuais do Ministério da Educação recomendam que se dê prioridade ao atendimento escolar integrado aos portadores de necessidades educativas especiais ^[?]. Com este projeto, pretende-se acelerar os métodos de aprendizagem e leitura, de um DV, por meio de caracteres produzidos por um computador levando assim a tecnologia na educação do aluno DV com custo acessível para a instituição de ensino e também para o seu uso particular, em casa, com a interação da própria família com o código Braille.

1.2 Procedimentos metodológicos - métodos e técnicas

O projeto proposto é composto de duas partes: *software* e *hardware*.

1.2.1 Software

O primeiro trabalho dentro deste projeto é o desenvolvimento de um *software* que tem a função de converter as informações de textos geradas por um microcomputador para o código Braille. Como conversão intermediária os textos são transformados no código ASCII. Os textos mostrados no monitor do computador podem ser de diversas naturezas, como por exemplo, textos gerados por processadores de textos ou similares. Um outro *software* (ou conjunto de *softwares*) embarcado em um microcontrolador, foi desenvolvido para a Interface Braille Automatizada, permitindo que esta possa ser utilizada em semáforos, prédios públicos, caixas eletrônicos, entre outras aplicações.

Como ferramenta de desenvolvimento, foi utilizado o *software* Visual Basic da Microsoft, cujo sistema será compatível com Microsoft Windows 2000, e Microsoft Windows XP.

Este *software* pode ser reescrito facilmente em outras linguagens de programação, a fim de que proporcione a sua utilização em outros sistemas operacionais,

como por exemplo o sistema operacional LINUX.

1.2.2 Hardware

A segunda parte desse projeto é o desenvolvimento de um sistema (*hardware*) para possibilitar à Interface Braille Automatizada comunicar dados ao DV. A Interface Braille Automatizada é composta de um conjunto de agulhas na forma matricial, (3 linhas por 2 colunas), sendo que a superfície será plana na ausência de comunicação do dado e que por comando, do caractere transformado em código Braille, produz o “alto relevo” correspondente àquela informação em código Braille. Neste caso, a matriz 3x2 de agulhas estará fazendo o papel de uma cela, ou celinha como é chamada pelos DVs. Um conjunto de celas, ou celinhas, poderá formar uma palavra ou mesmo uma frase. A velocidade de mudança para o próximo caractere dependerá do treinamento do DV e poderá ser controlada via *software* ou *hardware*.

Para a montagem do *hardware* foi escolhido o microcontrolador PIC 16F877 da Microchip, que será detalhado nos capítulos seguintes, os *softwares* MPLab para escrita do *software* embarcado no microcontrolador e o gravador ProPic 2 para gravação do componente.

1.3 Organização da dissertação

Este trabalho está dividido em cinco capítulos principais e a conclusão. Os capítulos são:

- Capítulo 1: Introdução
- Capítulo 2: A solução proposta: Revisão da literatura
- Capítulo 3: Descrição da Interface Braille Automatizada: Material e métodos
- Capítulo 4: Validação: Resultados obtidos
- Capítulo 5: Estado atual e perspectivas futuras: Discussão e sugestões para trabalhos futuros
- Conclusão

1.4 Resumo do Capítulo 1

Este capítulo é uma introdução sobre DVs, e sobre a necessidade de ferramentas para a realização de diversas atividades diárias do DV, principalmente a leitura.

Foram abordados objetivos e resultados esperados além de um breve resumo do *software* e *hardware* do projeto desenvolvido.

No capítulo seguinte, será mostrado com detalhes, o funcionamento do código Braille, a solução proposta, suas aplicações, vantagens e desvantagens.

2 *A solução proposta*

A solução proposta é o desenvolvimento de uma interface, denominada Interface Braille Automatizada, que irá automatizar celineha, a fim de produzir a escrita Braille que seja inteligível por um DV, e proporcionar uma melhoria na comunicação do mesmo, facilitando a sua inclusão social.

2.1 O código Braille

O Sistema Braille é um código universal de leitura tátil e de escrita usado por DVs.



Figura 2.1: Mãos tocando o papel e através do tato proporcionando a leitura.

A Figura ?? ilustra um DV lendo a partir do tato passando os dedos numa superfície de papel que contém os caracteres Braille.

2.1.1 Histórico do Braille

Há mais de 150 anos que o “Braille” é o meio usado por excelência pelos pacientes com perdas visuais severas para a leitura e escrita. Seu inventor é Louis Braille, foto na Figura ?. Devia ter pouco mais de quinze anos quando desenvolveu o seu código de escrita. O jovem francês, nascido em 1809 próximo de Paris, ficou cego aos três anos de idade, após um acidente, mas não desistiu de tentar aprender. Uma bolsa de estudo permitiu-lhe ingressar, em 1819, no Instituto para Jovens Cegos, em Paris, onde

se ensinava a ler através da impressão muito forte de textos em papel, que permitia dar relevo às letras. O sistema não era perfeito, mas possibilitava a leitura.

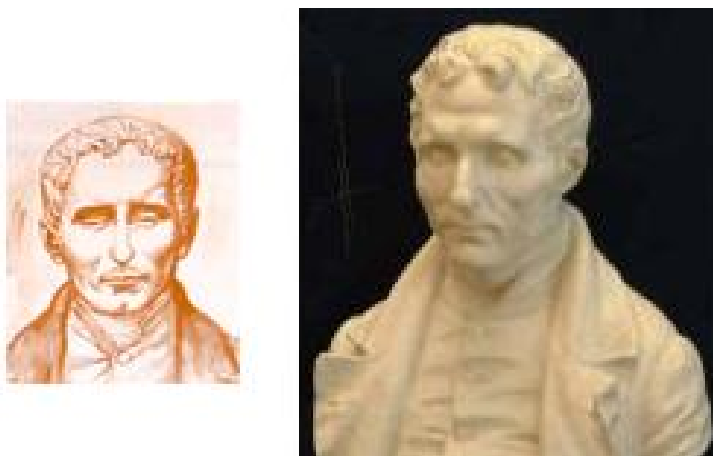


Figura 2.2: Louis Braille.

O maior problema era no momento de escrever, para o cego era impossível. Braille interessou-se então por um sistema de escrita, inventado pelo capitão Charles Barbier de La Serre, para transmissões noturnas em campanha de guerra, também baseado em pontos em relevo, e melhorou-o. Em 1829, publicou o primeiro manual onde o novo código que haveria de ficar para sempre com o seu nome aparecia sistematizado. Mas existem alguns documentos que provam que o jovem Louis já utilizava esse alfabeto há pelo menos cinco anos.

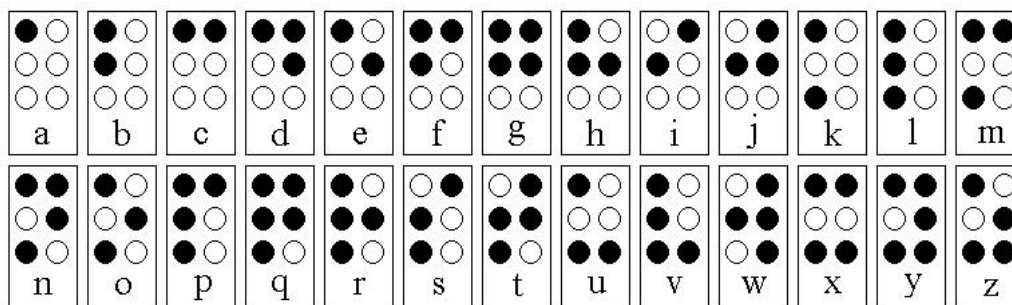


Figura 2.3: Código Braille, para representação do alfabeto. Cada conjunto de círculos maiores, que são em alto relevo, representa um caractere.

Nesta sua primeira versão do alfabeto Braille, o sistema estava praticamente definido - seis pontos em duas filas verticais de três pontos cada, num total de 63 sinais - mas haviam algumas combinações com traços que desapareceram oito anos depois, quando publicou a segunda versão da obra. A Figura ?? mostra o alfabeto, de 1837, que permaneceu praticamente inalterado até hoje. Louis Braille morreu em 1852 mas deixou um legado imprescindível para a população cega mundial. A sua vida e a sua obra podem ainda hoje ser descobertas no museu francês com o seu nome, onde, entre outros documentos, se encontram alguns dos primeiros textos escritos no novo alfabeto, na sua adolescência.

A Figura ?? mostra uma régua utilizada para se escrever em Braille, normalmente chamada de reglete. Esta régua é composta por uma prancheta que prende o papel, e uma régua que tem uma parte superior e uma parte inferior. Na parte superior, existe uma abertura para cada celinha e na parte inferior, o conjunto de seis pontos em declive. O papel é disposto entre estas duas partes. O escritor deve se utilizar de uma ferramenta tipo punção, e escrever da direita para a esquerda pressionando o punção de forma contrária à que se lê. Isto se deve ao fato de que para a leitura, deve-se virar o papel e sentir pelo tato as saliências causadas no papel pelo punção.



Figura 2.4: Reglete - Régua utilizada para a escrita Braille.

2.1.2 Funcionamento do Código Braille

A codificação Braille consiste em um conjunto de seis pontos, formados por duas colunas e três linhas, normalmente formando um retângulo de aproximadamente 6mm de altura por 2mm de largura. A esse conjunto normalmente é dado o nome de “celinha”. Algumas das várias combinações para a formação do alfabeto estão mostradas na Figura ??, onde os pontos de círculos na cor preta indicam o alto-relevo requerido para aquele específico caractere.

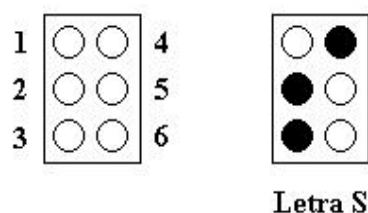


Figura 2.5: Nas celinhas os pontos são numerados de 1 a 6 e a combinação deles resulta no Código Braille.

Como pode ser visto na Figura ??, cada celinha, que possui 6 pontos, tem seus pontos numerados de 1 a 6, e a combinação deles representa um caractere, que neste exemplo nos mostra o caractere referente a letra “s”.

Existe um processo lógico para o entendimento da maioria dos caracteres representados pelo alfabeto Braille. Os caracteres foram divididos em sete grupos que possuem uma lógica de passagem de um grupo para o outro, exceto o grupo sete, como descrito na Tabela ??.

Tabela 2.1: Tabela de conjunto de caracteres e pontos

Grupo	Caracteres	Combinação dos pontos
1	“a”, “b”, “c”, “d”, “e”, “f”, “g”, “h”, “i” e “j”	1 2 4 5
2	“k”, “l”, “m”, “n”, “o”, “p”, “q”, “r”, “s” e “t”	grupo 1 + ponto 3
3	“u”, “v”, “x”, “y”, “z”, “ç”, “é”, “á”, “è” e “ú”	grupo 2 + ponto 6
4	“â”, “ê”, “ï”, “ô”, “ù”, “à”, “ï”, “ü”, “õ” e “w”	grupo 3 - ponto 3
5	“í”, “ã”, “ó”, “,”, “;”, “!”, “*”, “\$”, “.” e “?”	2 3 4 5 6
6	“1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9” e “0”	3 4 5 6 + grupo 1
7	sinais	não seguem grupos de pontos

A Tabela ?? pode ser melhor entendida nos tópicos abaixo:

- O primeiro grupo, representado na Figura ??, é a combinação dos pontos 1, 2, 4 e 5 que formam as letras de “a” a “j”.

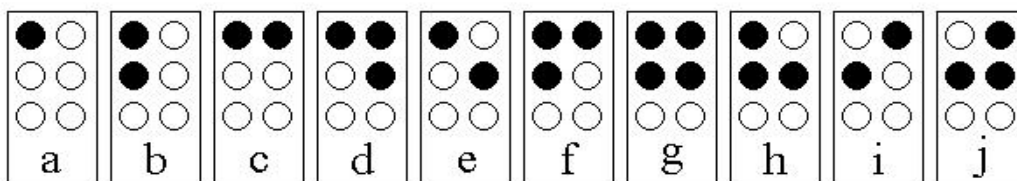


Figura 2.6: A combinação dos pontos 1, 2, 4 e 5 formam o grupo de letras de “a” a “j” .

- O segundo grupo, representado na Figura ??, acrescenta o ponto 3 ao grupo anterior e forma assim uma nova combinação resultando um novo grupo de letras de “k” a “t”.

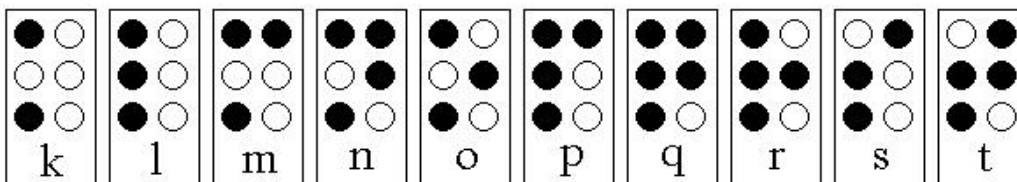


Figura 2.7: Acrescentando-se o ponto 3 ao grupo de letras de “a” a “j” é formado o grupo das letras de “k” a “t”.

- O terceiro grupo, representado pela Figura ??, acrescenta o ponto 6 ao grupo anterior resultando assim num novo grupo, as letras de “u” a “ú”.

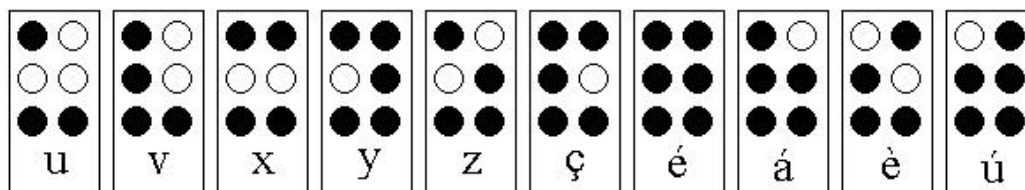


Figura 2.8: Acrescentando-se o ponto 6 ao grupo de letras de “k” a “t” é formado o grupo das letras de “u” a “ú”.

- O quarto grupo, representado pela Figura ??, apenas retira o ponto 3 do grupo anterior e forma assim o novo grupo das letras de “â” a “w”.

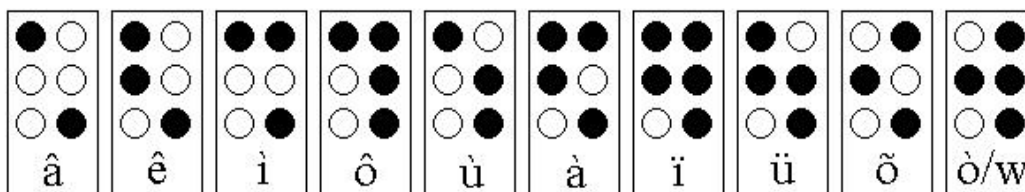


Figura 2.9: Retira-se o ponto 3 do terceiro grupo das letras de “u” a “ú” e forma-se o grupo das letras de “â” a “w”.

- O quinto grupo, representado pela Figura ??, inclui agora símbolos e não segue mais a mudança de um grupo para outro. Este grupo é resultado da combinação dos pontos 2, 3, 4, 5 e 6 e é o conjunto de caracteres de “í” a “?”.

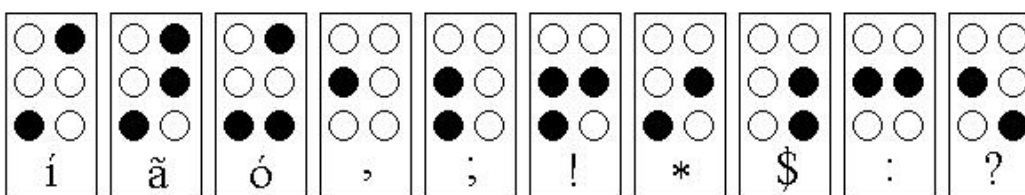


Figura 2.10: A combinação dos pontos 2, 3, 4, 5 e 6 formam o grupo de caracteres de “í” a “?”.

- O sexto grupo refere-se ao grupo do números. Agora começam os caracteres Braille que utilizam mais de uma celinha para compor um único caractere. Por exemplo, as letras maiúsculas são usadas com recurso de um prefixo, pontos 4 e 6 conforme a Figura ??, em uma celinha, seguido da letra desejada na outra celinha. Assim, nesta lógica de utilização de duas celinhas, os números são compostos do primeiro grupo, de “a” a “j” na segunda celinha, porém com um caracter especial na primeira celinha como descrito na Figura ??.

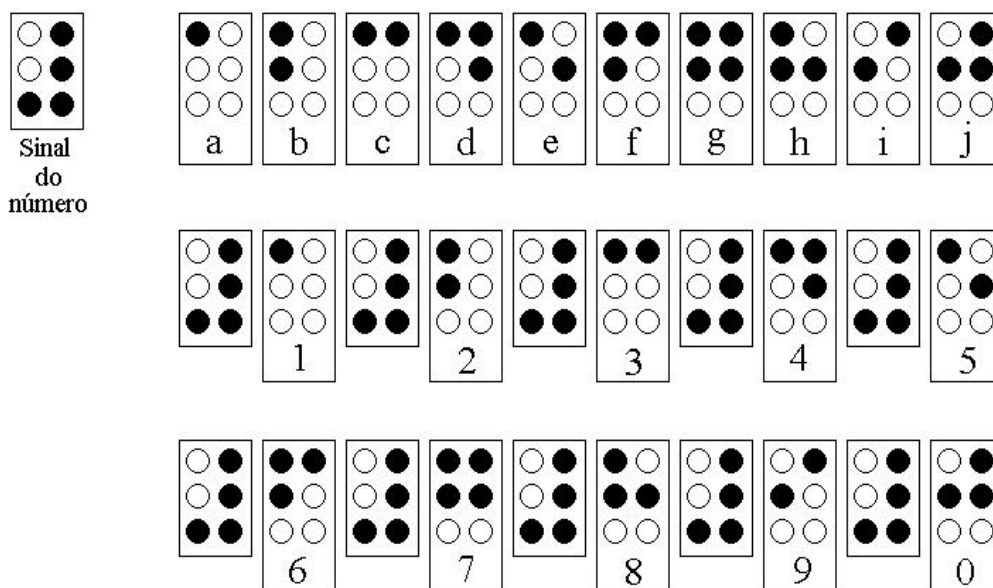


Figura 2.11: Os números são formados pela introdução de um sinal especial e as letras de “a” a “j”.

- A Figura ??, refere-se ao sétimo e último grupo. Este é o grupo dos caracteres de sinais e não há uma lógica na sua composição na seqüência de um grupo para outro. É necessário que, com estudos e prática, se memorize os códigos destes caracteres.

2.2 Interface Braille Automatizada

Com o intuito de solucionar os problemas encontrados no dia-a-dia de um DV é que desenvolvemos a Interface Braille Automatizada. O funcionamento da presente inovação se dá da seguinte forma: um *software* transforma textos gerados, de um sistema computacional, para uma codificação. Esta codificação é enviada a um *hardware*, que aqui denominamos de Interface Braille Automatizada. Esta interface é responsável pela automatização das celinhas, que produz em sua superfície, inicialmente plana, o código Braille necessário para a comunicação com o DV. Desta forma, estaremos automatizando a celinha, isto é, transformando, uma superfície inicialmente plana, em caracteres codificados em Braille.

A IBA é composta de *hardware* e de *software*, como já mencionado. O *hardware* foi dividido em alguns módulos para facilitar o projeto e melhorar a compreensão do mesmo. A celinha automatizada é apenas uma pequena parte de todo o circuito necessário para o funcionamento de todo o sistema como pode ser visto na Figura ??, onde nota-se os pontos da celinha.

A Figura ?? mostra um esboço do que é a celinha da Interface Braille Automatizada. Quando todos os pontos estiverem em repouso, não indica símbolo algum,

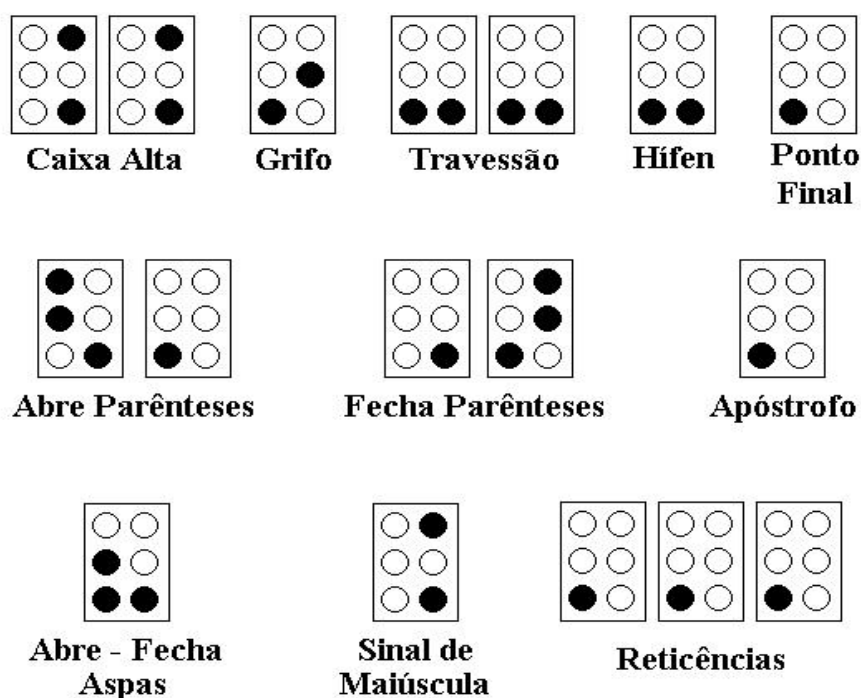


Figura 2.12: Grupo de sinais.

porém, como mostrado na Figura ??, dois pontos da celinha estão ativos, neste caso simbolizando o caractere “c”.

2.2.1 Aplicações do Sistema Proposto

- Uso do computador: O “SoftBraille”, armazenado no computador, converte os textos gerados por processadores de texto ou páginas de internet em uma codificação apropriada, armazenando em um arquivo texto sem formatação. Este arquivo é enviado para a IBA (Interface Braille Automatizada). A IBA armazena, converte e escreve a informação em código Braille. Opções de repetições de caracteres, frases ou parágrafos, armazenamento de arquivos, são parte integrante do sistema. A Figura ?? ilustra o DV utilizando o microcomputador pessoal, para leitura de páginas de internet e leitura de textos (cartas, livros, documentos e assemelhados). O DV percebe, através do tato, a informação codificada.
- Informações Relevantes: A IBA é um meio de comunicação, para informar o DV, das características de instalações públicas, de museus, galerias ou assemelhados. Pode ser instalada estrategicamente em prédios públicos (museus, galerias entre outros) para fornecer informações dinâmicas e relevantes aos DVs. A Figura ?? ilustra a utilização da IBA para este caso.
- Caixa eletrônico e Agências Bancárias: os caixas eletrônicos poderão ser equipados com a IBA para informar ao DV sua movimentação bancária ou mesmo proceder algum tipo de operação bancária como saques e depósitos. Através da IBA o DV

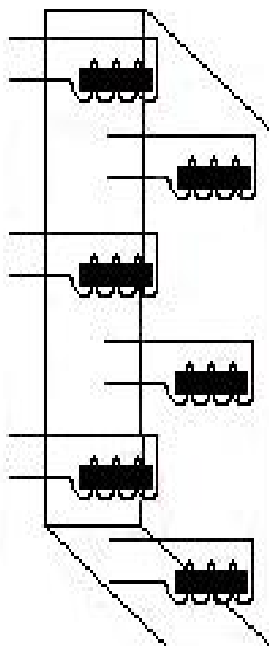


Figura 2.13: Celinha e as bobinas que acionarão os pontos do Alfabeto Braille.

poderá verificar se o que ele digitou no teclado está correto ou não. Em terminais de uso exclusivo para DV, o monitor de vídeo poderá ser dispensado. A Figura ?? ilustra o DV utilizando um terminal bancário.

- Alfabetização de um DV: através de áudio, em conjunto com a Interface Braille Automatizada, o “SoftBraille” é direcionado para agilizar a alfabetização de DV. A Figura ?? mostra um DV sendo alfabetizado.

Uma vez que a Interface Braille Automatizada estiver conectada a um microcomputador, através do SoftBraille, o DV pode digitar um caractere, ouvir o som deste mesmo caractere e passar o dedo na Interface Braille Automatizada e sentir através do tato a informação codificada em Braille na celinha automatizada.

- Sinalização nas ruas (semáforos e informações relevantes ao pedestre): A Interface Braille Automatizada interligada a semáforos das cidades, onde através do tato, e sinais gerados pela IBA, o DV poderá receber a mensagem das condições do tráfego, além de outras informações. Através de sistemas computacionais, cada semáforo poderá ter uma IBA que estará continuamente informando ao DV (via código Braille) as condições de segurança (sinal vermelho ou verde). Poderão ser colocadas interfaces em ponto de ônibus (ou assemelhados como estações de trens, metrô e rodoviárias) para informações do percurso, linhas e horários. A Figura ?? apresenta algumas sinalizações de rua, como semáforos, ponto de ônibus e hora certa.

Desta forma, o DV com mais informações, poderá andar com mais segurança nas

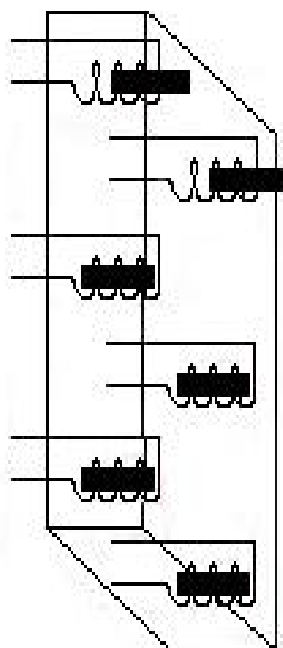


Figura 2.14: Celineha e as bobinas acionadas nos pontos do Alfabeto Braille que representam o caractere “c”.

ruas e conseguir se deslocar mais facilmente com informações sobre linhas de ônibus e outras informações relevantes em tempo real.

2.2.2 Vantagens do Sistema Proposto

Durante entrevistas com DVs, ficou claro nesse momento que a maior vantagem do projeto, é a possibilidade de alfabetização do DV por si próprio, além de, ao mesmo tempo, ele poder treinar a digitação no teclado do microcomputador. Além disso, com as demais possibilidades de aplicação descritas em 2.2.1, pode-se ter uma inclusão social muito grande. Não somente o DV poderá ter acesso ao código Braille, mas toda a sua família e os que convivem com ele.

Uma outra grande vantagem é o incentivo que o governo está dispondo para o desenvolvimento de equipamentos ligados a saúde e melhoria na qualidade de vida de deficientes.

2.2.3 Desvantagens

Entre elas pode-se citar que, devido à diferença da textura do relevo em relação a uma folha de papel, pode haver uma demora de adaptação.

Poderá haver uma certa rejeição inicial, pois a sensibilidade do tato ao toque

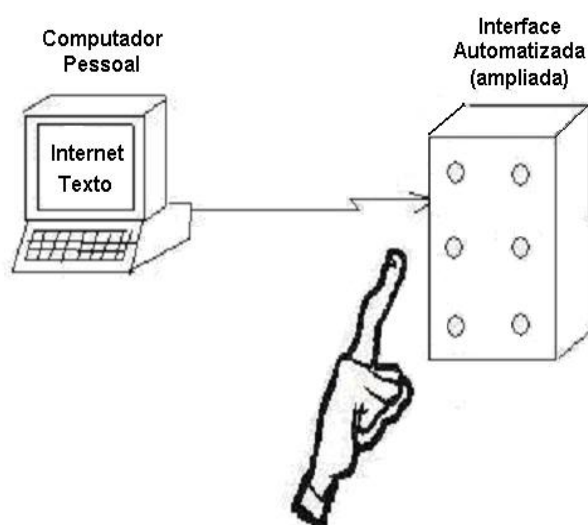


Figura 2.15: O DV poderá ler documentos gerados por processadores de texto ou páginas de internet.

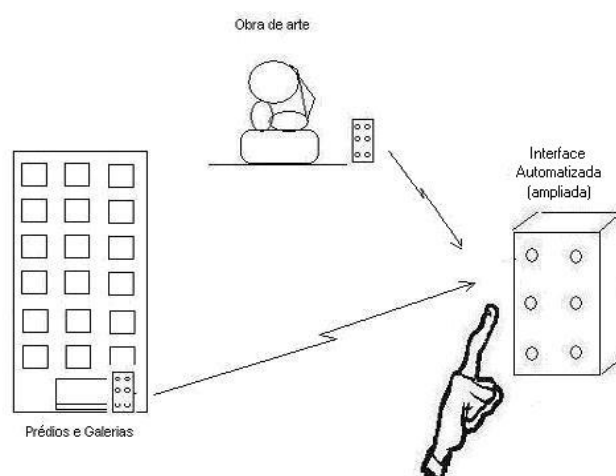


Figura 2.16: Interface Braille Automatizada usada como meio de comunicação para informações relevantes.

em uma folha de papel em relevo é diferente dos pinos que fazem o relevo na IBA, mas acredita-se ser de fácil transposição.

2.3 Resumo do Capítulo 2

Neste capítulo pode ser visto com detalhes, o funcionamento do código Braille, a solução proposta, suas aplicações, e algumas vantagens e desvantagens.

No capítulo seguinte, veremos a descrição da Interface Braille Automatizada proposta, e seus componentes de *Hardware* e *Software*.

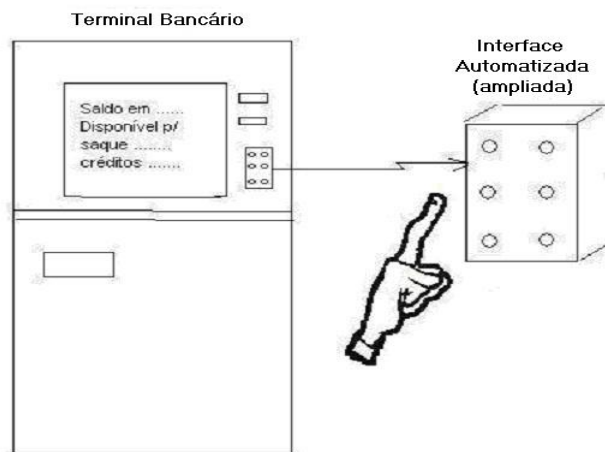


Figura 2.17: Interface Braille Automatizada ligada a um caixa eletrônico.

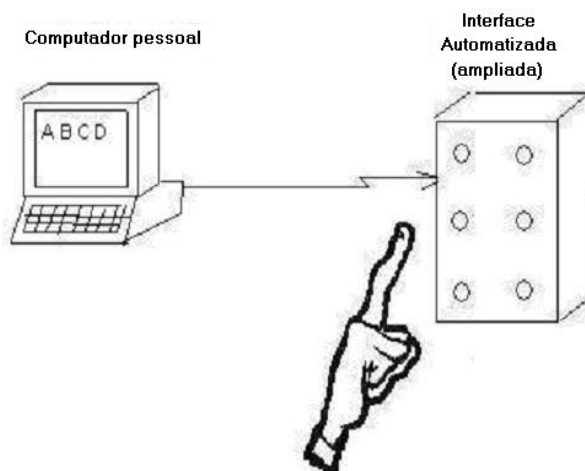


Figura 2.18: Interface Braille Automatizada ligada a um microcomputador destinada a alfabetização de um DV.

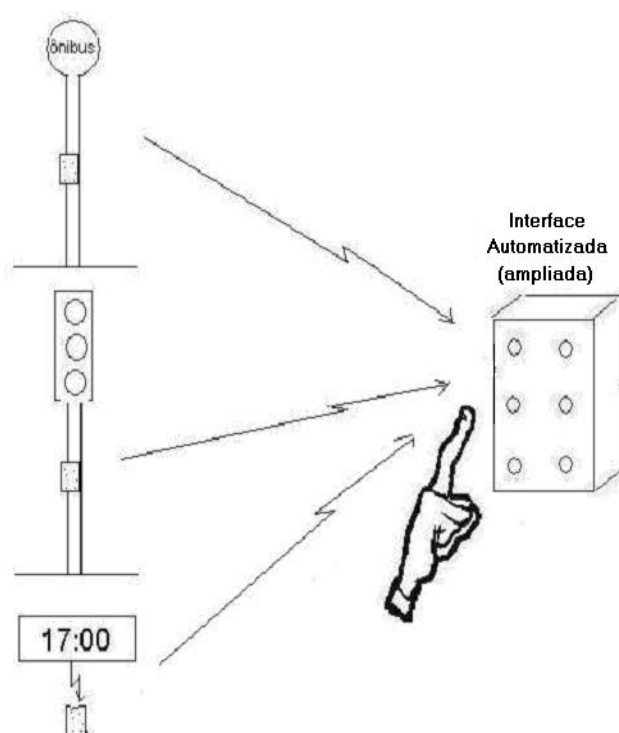


Figura 2.19: Outros exemplos de Interface Braille Automatizada ligada independente do microcomputador.

3 *Descrição da Interface Braille Automatizada*

A presente inovação trata-se de uma interface denominada Interface Braille Automatizada composta por *software* e *hardware* que permite ao DV interpretar os textos (caracteres produzidos), através da automatização das celinhas, que representam em relevo, um caractere no alfabeto Braille. A Figura ??, mostra o conceito geral de funcionamento e acionamento do que definimos como Interface Braille Automatizada.

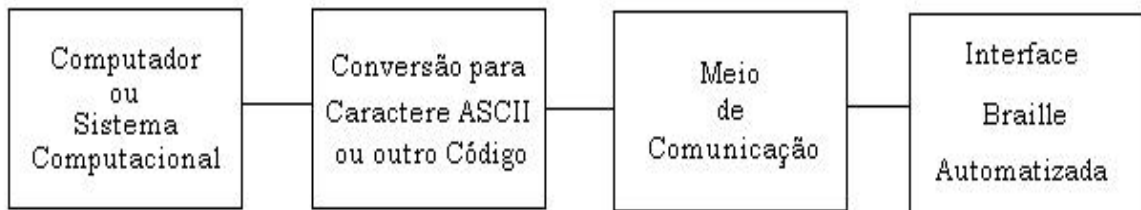


Figura 3.1: Diagrama em blocos do conceito geral de funcionamento da Interface Braille Automatizada.

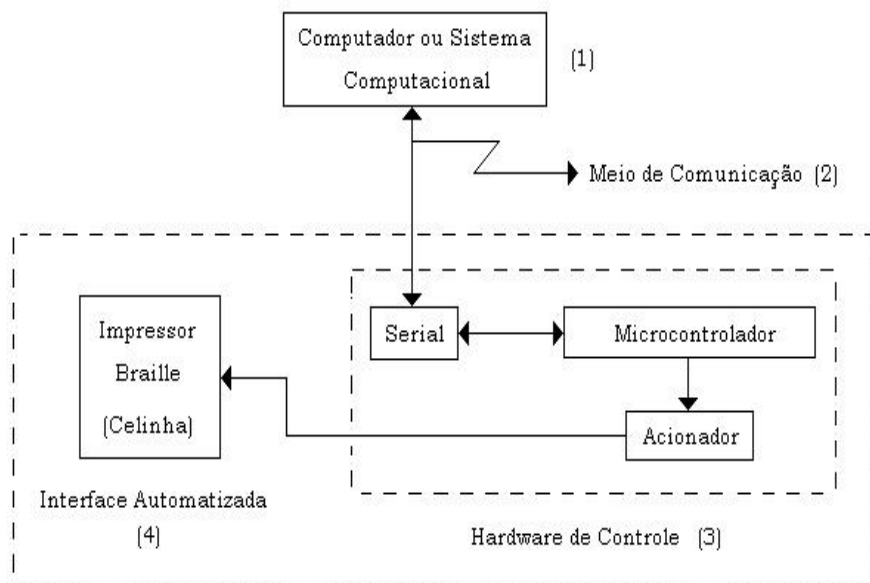


Figura 3.2: Expansão em blocos de todo o sistema.

Para uma melhor compreensão do funcionamento de todo o sistema, a Figura ?? mostra a expansão em blocos, facilitando assim o entendimento dos módulos:

- (1) Computador ou sistema computacional que faz a conversão para caractere ASCII ou outro código.
- (2) O meio de comunicação.
- (3) O *hardware* de controle da Interface Braille Automatizada.
- (4) Os módulos que constituirão as várias celinhas possíveis no módulo da Interface Automatizada.

A Interface Braille Automatizada é externa ao sistema computacional sendo composta de um “*hardware* de controle” e de uma “interface automatizada”. O *hardware* de controle possui um *software* próprio e embarcado na unidade, gravado no microcontrolador, denominado de *software* de controle. A interface automatizada é propriamente o contato físico com o DV, que através do tato ele fará a leitura em código Braille.

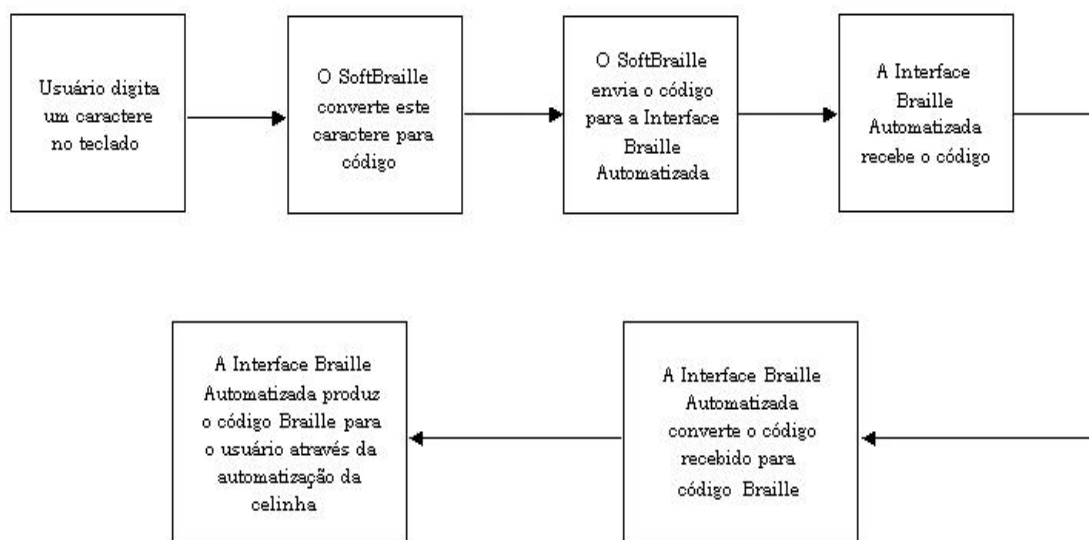


Figura 3.3: Diagrama em blocos da interação entre o SoftBraille (presente no sistema computacional) e a Interface Braille Automatizada.

Está também prevista, a possibilidade de se utilizar um conjunto modular de interfaces automatizadas, para a formação de palavras ou mesmo frases, adequando assim o equipamento para várias situações que poderão surgir. A Figura ??, mostra um dos exemplos da utilização do *software*, denominado de “SoftBraille”. Neste exemplo, o sistema foi direcionado para executar a tarefa de alfabetização de um DV. O *software* também tem, entre outras funções, a finalidade da comunicação entre o computador e a IBA, que por sua vez provoca a alteração de sua superfície conforme o código do caractere.

3.1 Software

O *software*, cujo nome foi escolhido SoftBraille, tem a função de transformar os textos nos seus vários formatos de origem, gerados em um computador ou em um sistema computacional, em código ASCII. Este código é enviado ao *hardware* e este transforma a informação em código Braille, possível de ser lida na interface automatizada, e que também deve ser lido por um DV.

As telas e o manual do programa estão descritos no Apêndice C.

3.1.1 Desenvolvimento do Software SoftBraille

Para o desenvolvimento do *software* foi utilizada a ferramenta de desenvolvimento Visual Basic 6.0 da Microsoft. Este ambiente de programação foi escolhido pelo fato de ser encontrada uma vasta quantidade de rotinas prontas desse ambiente de programação na internet.

As rotinas principais do programa podem ser encontradas no Apêndice A. Lá pode-se notar rotinas como o tratamento da porta serial, rotinas que executam os sons e tratamento dos textos. A seguir tem-se breves comentários das principais rotinas.

3.1.1.1 Rotina da porta serial (MSComm)

O MSComm é um recurso intrínseco ao Visual Basic e portanto bastante simples de ser utilizado. No caso, tem-se a porta serial em questão. Foi fixada como COM1 (MSComm1.CommPort = 1). Com a opção settings, do MSComm, foi determinada a velocidade da porta e controle de erro e paridade; neste caso, sem necessidade de velocidade, foi determinado como 2.400 KBps e sem paridade. A opção inputLen=0, ainda do MSComm, diz que a porta não está selecionada para receber nada pela serial.

3.1.1.2 Rotina que executa os sons (PlaySound)

Outra rotina interessante é a que executa o som dos caracteres digitados no teclado do microcomputador pelo usuário. Para cada caractere do teclado existe um arquivo com extensão .WAV associado¹

Durante a execução do programa, a qualquer instante a função PlaySound pode ser chamada para executar o som da tecla pressionada. Para isso, basta informar o caminho que está gravado o som relativo ao que se deseja (PlaySound "c:\SoftBraille\sons\space.WAV")

¹Teclas de funções especiais como F1, F2 ou ESC e ENTER não foram implementadas com sons.

3.1.1.3 Alfabetiza

Com esta rotina, tem-se a intenção de alfabetizar e assim, a cada caractere digitado, o DV ouve o que digitou, e o *software* envia para a interface este mesmo caractere no padrão ASCII, e este aciona a interface fixa para que esta movimente os solenóides a fim de que o DV sinta pelo tato o mesmo caractere codificado para Braille em alto relevo na IBA. Dessa forma, o DV não só aprende a digitar no teclado de seu microcomputador, como também alfabetiza-se² em Braille.

3.1.1.4 Capturando a tecla pressionada (KeyDown)

Para capturar o que foi digitado no teclado, foi utilizado o comando KeyDown do Visual Basic. O comando KeyDown captura o valor digitado em ASCII e coloca em um *label*. O label foi utilizado para facilitar na programação, pois a cada caractere digitado, o ASCII foi apresentado na tela. Para a execução do programa, esta informação pode ser útil para orientar os videntes que estejam utilizando o programa e que estejam executando alguma manutenção no sistema.

3.1.1.5 Conversão de arquivos

Com esta função, faz-se a conversão dos arquivos para a extensão .TXT, livre de figuras e de formatações. Também são retirados os espaços em branco repetidos e linhas em branco. Esta função, pela complexidade e atenção exigida foi colocada em um módulo separado da rotina principal que é chamada durante a execução do programa.

Quando o DV digita o caminho do arquivo que ele deseja trabalhar, esta função automaticamente retira toda a formatação do arquivo, transformando-o em um arquivo com extensão .TXT , e dessa forma, passível de ser codificado para ASCII e posteriormente para Braille.

Para a conversão dos arquivos, foram escolhidas as extensões .DOC, .DOT, .RTF, .HTM, .XLS e .TXT.

Exemplo: O DV recebe um disquete com um arquivo chamado AULA.DOC de um colega da faculdade. Chegando em casa, ele coloca o disquete no drive A: de seu microcomputador e executa o SoftBraille. Após uma breve mensagem audível de saudação, o *software* fica aguardando que o DV digite o caminho do arquivo que ele deseja trabalhar. Portanto, ele digita A:\AULA.DOC e pressiona ENTER. O softBraille irá então converter o arquivo para AULA.TXT e irá salvar automaticamente este arquivo em uma pasta apropriada do sistema. Com isto, todas as formatações do texto foram excluídas e o arquivo está salvo e pronto para ser utilizado na IBA.

²O termo alfabetiza refere-se ao contato do DV com o alfabeto Braille.

3.2 Hardware

Para a automatização de uma celinha, podem ser utilizadas várias técnicas. Dentre as possibilidades podemos citar a deformação piezoelétrica, onde cada ponto altera sua superfície, com a excitação de um cristal piezoelétrico, resultando em deformação elástica ou vibração. O protótipo proposto e montado da Interface Automatizada utiliza um conjunto de seis bobinas, uma para cada ponto da celinha. As bobinas são interligadas a um sistema de codificação, gerada pela Interface Braille Automatizada, que converte os caracteres provindos de um computador, ou por um sistema computacional, executando o SoftBraille, para o código Braille. Esta conversão de código aciona as bobinas, que fazem parte da constituição dos caracteres, resultando uma superfície em alto-relevo. A superfície formada representará o código Braille gerado por um computador, ou sistema computacional.

Na ausência de qualquer informação a superfície será plana. Esta situação é ilustrada na Figura ??.

O tamanho físico da Interface Automatizada deverá ter as mesmas dimensões adotadas para as “celinhas” padrões utilizadas mundialmente.

A Figura ??, que representa uma Interface Automatizada, ilustra o acionamento de dois pontos, sendo neste caso acionados por duas bobinas. Nesta condição a superfície apresentará alto relevo. Ao passar os dedos sobre esta superfície, o DV poderá ler o caractere codificado em Braille. Analizando-se a ilustração da Figura ??, duas bobinas foram alimentadas com energia suficiente para fazer com que dois pinos se deslocassem para cima. A combinação do alto relevo formado, para este exemplo, representa a letra “c” do alfabeto.

Poderá ser montado um conjunto de “celinhas” com o propósito de formar palavras inteiras, ou mesmo frases. Ao comando do DV, poderá se obter repetições de caracteres, palavras ou frases.

3.2.1 Desenvolvimento do Hardware

O *Hardware* foi inicialmente montado em um *ProtoBoard* como ilustrado na Figura ?. Cada conjunto de seis LEDs corresponde a uma celinha, onde cada LED é um ponto da mesma. A idéia é que se é possível representar o Braille através dos LEDs, basta implementar um conjunto eletro-mecânico ou piezoelétrico capaz de alterar a superfície plana a fim de que o DV possa fazer a leitura. Após algumas alterações no circuito inicial foi projetada a primeira placa de testes, onde ficam o microcontrolador, o *drive* de corrente, e o *Latch* que faltava no projeto inicial. A Figura ?? ilustra o desenho das trilhas desta placa. Posteriormente foi desenvolvida a segunda placa, para ampliação de duas para oito celinhas no sistema. O desenho das trilhas desta segunda placa é ilustrada na Figura ??³. Obtido o desenho, foi desenvolvido todo o processo de confecção das placas.

³As ilustrações destas placas não estão em tamanho real.

Em ambas as placas foram colocados conectores do tipo IDC-M de dezesseis pinos que tem a função de fazer a interligação entre as placas CPU e de expansão. Outro conector IDC-M foi colocado na placa da CPU, Figura ??, para fazer a interligação das teclas de controle do *Hardware*. O circuito foi dividido em partes menores, ou módulos, para facilitar o entendimento. O circuito elétrico será explicado na subseção “Funcionamento do *Hardware*”. As Figuras ?? e ?? ilustram as placas de controle e de expansão montadas com os componentes.

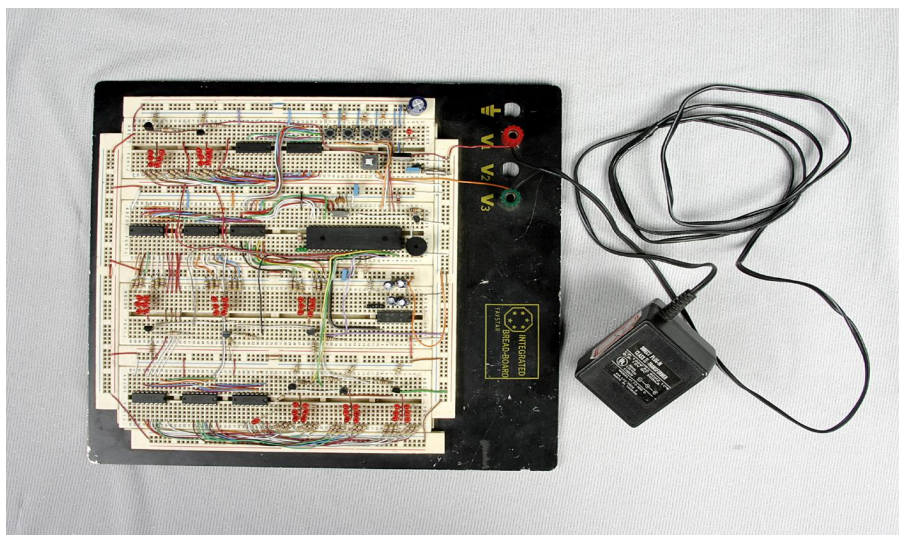


Figura 3.4: Primeiro circuito montado em *protoboard* com oito conjuntos de seis LEDs representando oito celinhas.

3.2.2 A escolha do Microcontrolador

Muitos são os microcontroladores disponíveis no mercado e o escolhido foi o microcontrolador PIC 16F877 da Microchip. O conjunto de motivos pela escolha foi: o preço; o fato de ter uma boa quantidade de memória disponível; e também o fato de haver um ambiente de desenvolvimento gratuito chamado MPLAB, da própria Microchip⁴. Para este projeto, a velocidade não é foco, portanto um oscilador de 20MHz é o suficiente. Para a gravação do microcontrolador, foi utilizado a interface Propic2, também da Microchip.

Este microcontrolador possui arquitetura RISC, com barramento de 14 Bits e set primário de 35 instruções, como exemplo: BSF (bit set file), BCF (limpa o bit do arquivo), GOTO (ir para), CALL (chamada), RETURN (retorno). Memória interna (RAM) de 368 Bytes. Memória de programa de 8.000 instruções de 14 Bits. Esta memória é onde o *Software* embarcado reside. Possui EEPROM interna de 256 Bytes, 15 tipos de interrupções disponíveis, e com isso pode-se gerar interrupções pelos periféricos como pela I/O, INT, ou *Timer* entre outros não utilizados neste projeto. O ciclo de instrução é de 200ns com um *clock* externo de 20MHz.

⁴Disponível em www.microchip.com

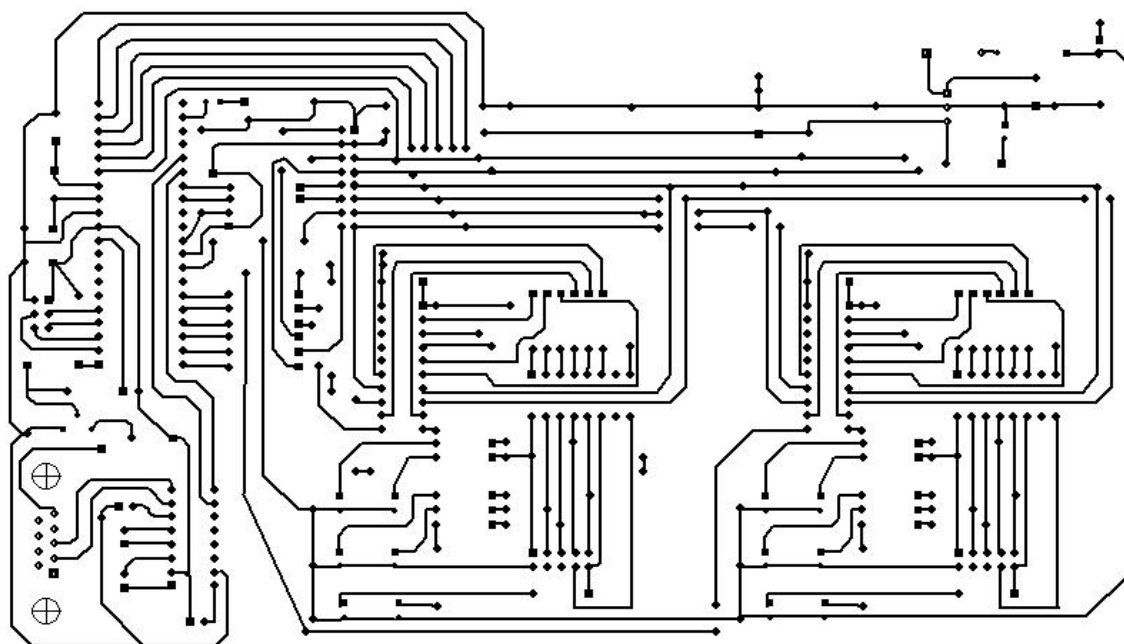


Figura 3.5: Desenho das trilhas da placa experimental de controle da Interface Braille Automatizada.

Outra grande facilidade encontrada neste componente é a grande quantidade de I/O bidirecionais disponíveis. Estas I/O são as portas de 8 Bits RA, RB, RC, RD e RE, conforme descrito na Tabela 3.2.

3.2.3 O Software Embarcado

O *software* embarcado, é o *software* desenvolvido para o microcontrolador. Este *software* define o funcionamento de todo o *hardware*.

Ele executa a conversão dos caracteres ASCII provenientes da porta serial, para Braille, e controla o barramento para que a celinha automatizada possa reproduzir o caractere desejado.

As rotinas principais do programa podem ser encontradas no Apêndice B.

3.2.4 Funcionamento do Hardware

3.2.4.1 A Fonte de Alimentação

Para o projeto, está sendo utilizada uma fonte de alimentação comum de 12V/5A. Porém, a alimentação do microcontrolador e dos CIs 74HC373 é de 5V. Assim, um segundo estágio de fonte é utilizado como na Figura ???. Um fusível foi colocado em série para proteger o circuito. Com a mesma finalidade de proteção foi introduzido um diodo retificador 1N4007 em série com o fusível. Este diodo evita danos ao circuito,

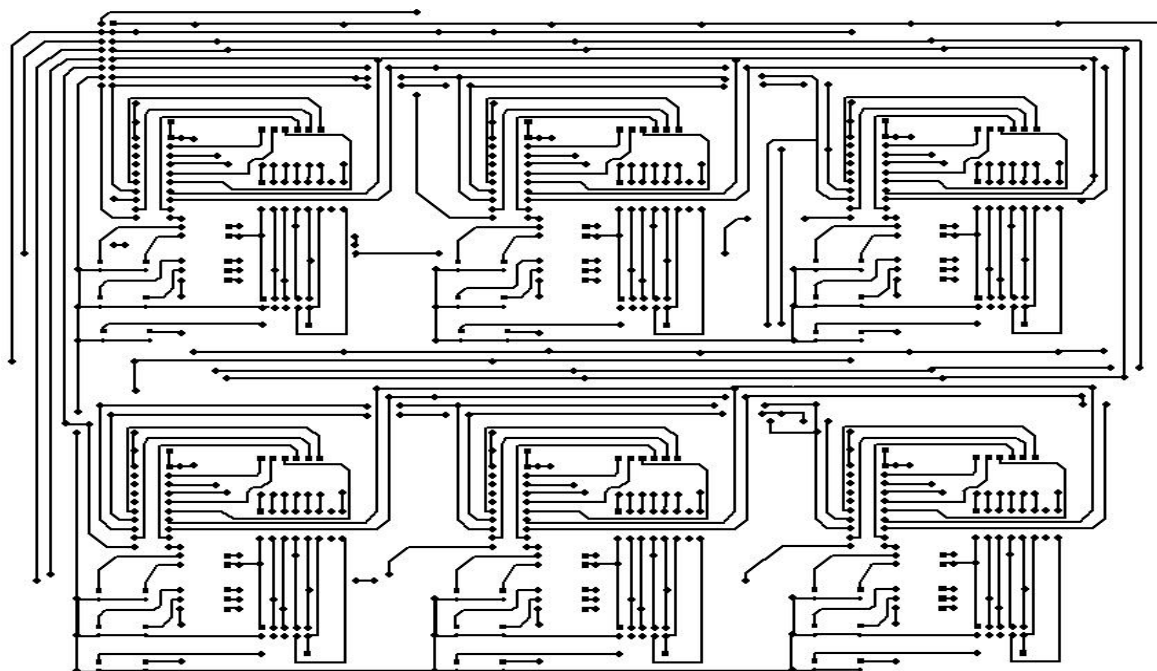


Figura 3.6: Desenho das trilhas da placa experimental de expansão de celinhas da Interface Braille Automatizada.

caso seja ligado a uma fonte de alimentação com polaridade invertida. A saída VCC2 é a alimentação de 5V para o circuito, e a saída VCC1 é a alimentação 12V direta para acionamento dos reles e LEDs representando as celinhas.

O capacitor de 100nF foi colocado para ajudar a retirar interferências e ruídos da fonte. O capacitor de 470 uF tem a função de manter estável o nível de saída da fonte.

O CI 7805 é um regulador de tensão com nível de saída de 5V. O LED foi colocado apenas para verificar que o equipamento está ligado e o resistor de 390R limita a corrente para evitar a sobrecorrente no LED.

3.2.4.2 A Comunicação Serial

A porta serial é a interligação entre o microcomputador e a Interface Braille Automatizada. Para tanto é utilizado o CI RS 232 conforme mostra a Figura ??.

A Figura ?? ilustra o circuito serial utilizado no projeto. Os pinos associados aos capacitores estão relacionados ao circuito elevador e inversor de tensão utilizada na conversão do sinal TTL com nível de 5V para RS-232 de +7,5 a +15 e de -7,5 a -15 V. Este circuito é chamado de *Charge Pump*. A Tabela ?? ilustra uma descrição dos terminais do circuito integrado RS232 utilizado no projeto e representado na Figura ??.

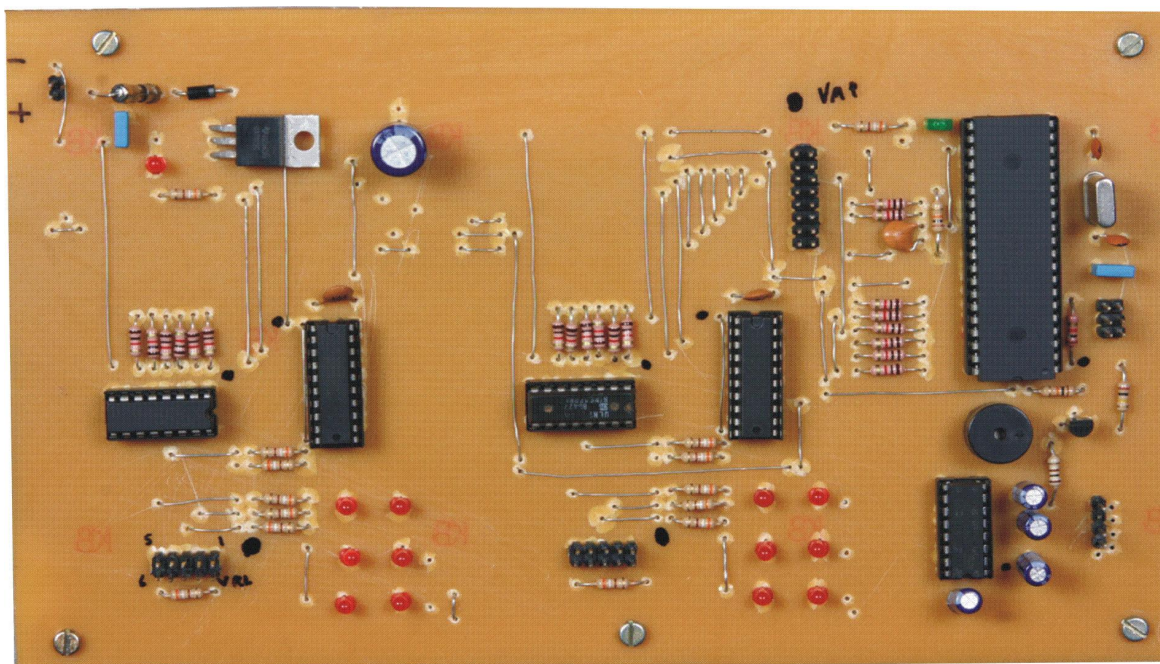


Figura 3.7: Placa experimental de controle da Interface Braille Automatizada montada com os componentes.

3.2.4.3 Módulo de controle

A Figura ?? representa o microcontrolador e seus terminais com a nomenclatura utilizada neste projeto. A Tabela ?? detalha melhor o componente e seus terminais.

A Figura ?? mostra o cristal piezoelétrico ligado aos terminais 13 e 14 do microcontrolador. Este cristal foi escolhido como sendo de 20 MHz, pois é a taxa de *clock* máxima para este microcontrolador, explorando assim o máximo de seu potencial. Para se obter estabilidade no oscilador, colocou-se dois capacitores de 18pF entre cada terminal do cristal e o terra, recomendado pelo próprio fabricante ^[?]. Outro capacitor, (C5), foi colocado entre a alimentação do microcontrolador e o terra, para evitar interferências, chamado de capacitor *By Pass*.

No pino 1 do microcontrolador, foi colocado um resistor de 1K, entre este terminal e a alimentação VCC2, para limitar a corrente. Este pino é o “*Master Clear*”, que com nível baixo gera o *reset* do microcontrolador, e com VCC2 ele funciona normalmente, por isso foi colocado o resistor, ou seja para manter o nível alto neste pino. No momento da gravação este pino deve ser levado à tensão de 13V para a gravação.

As portas RA, RB, RC, RD e RE foram quase em sua totalidade utilizadas como I/O digitais com exceção da porta RC que foi utilizado o RC6 e RC7 para comunicação serial.

De RA0 a RA5, tem-se uma porta bidirecional de 8 Bits, embora externamente apresenta apenas 6 Bits. Esta porta está sendo utilizada de RA0 a RA3, como

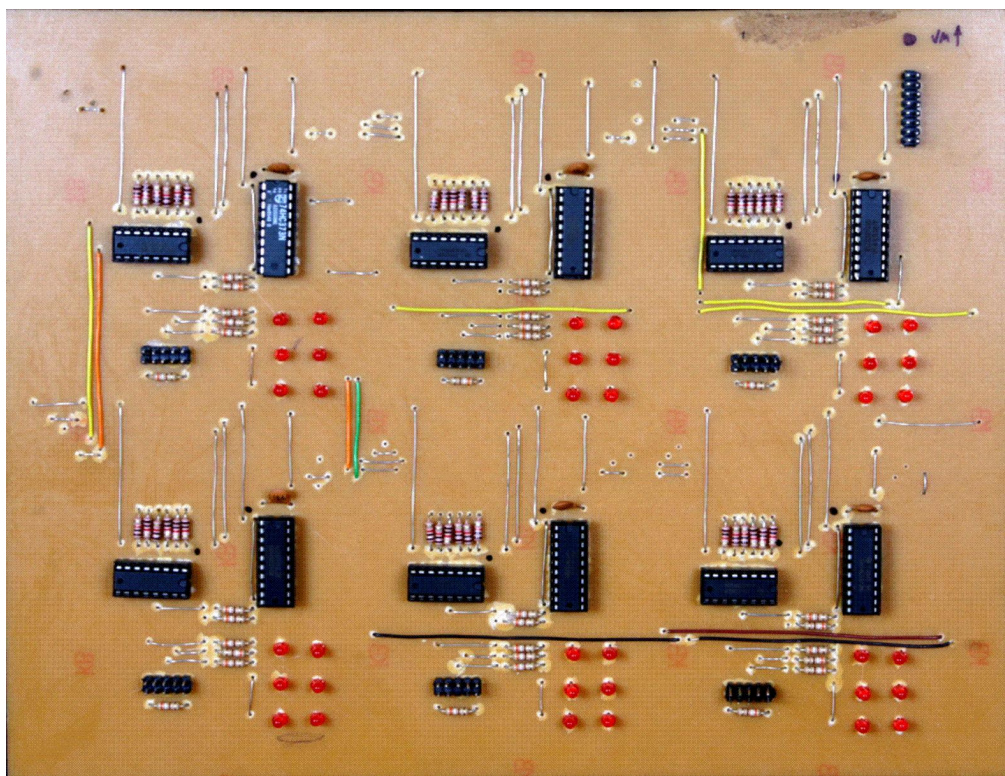


Figura 3.8: Placa experimental de expansão de celinhas da Interface Braille Automatizada montada com os componentes.

portas de entrada ligadas ao circuito do teclado de controle do *hardware*. Este teclado será melhor detalhado adiante. As portas RA4 e RA5 não estão sendo utilizadas e estão reservadas para futuras ampliações de celinhas ou outras necessidades de *hardware*.

O resistor do pino 33, de $10k\Omega$, mantém a entrada RB0 em nível alto. Esta porta pode ser utilizada para gerar interrupções no microcontrolador para executar rotinas específicas com chamadas de *hardware* externo. Neste projeto, não está sendo utilizado esta facilidade, e assim está sempre em nível alto de sinal. De RB0 a RB7, tem-se outra porta bidirecional de 8 Bits com recurso de interrupção na porta RB0, como já descrito. Neste projeto, as portas RB2 a RB7 foram utilizadas como saída dos sinais CELA1 a CELA6, que vão acionar as respectivas celinhas. A porta RB1 não está sendo utilizada e está reservada para futuras ampliações de celinhas ou outras necessidades de *hardware*.

Outra porta bidirecional é a porta RC. De RC0 a RC3 foi configurado como sendo os pontos 1 a 4 do barramento paralelo dos pontos das celinhas. RC4 e RC5 não estão sendo utilizadas e estão reservadas para futuras ampliações de celinhas ou outras necessidades de *hardware*. Como comentado, RC6 e RC7 foram utilizados para a comunicação serial.

De RD0 a RD7, tem-se outra porta bidirecional de 8 Btis configurável como entrada ou saída. Neste projeto, as portas RD3 e RD5 foram utilizadas como saída dos sinais CELA7 e CELA8 respectivamente, que vão acionar as respectivas celinhas. A porta RD4 foi utilizada como saída e envia um sinal pulsante que faz com que o LED

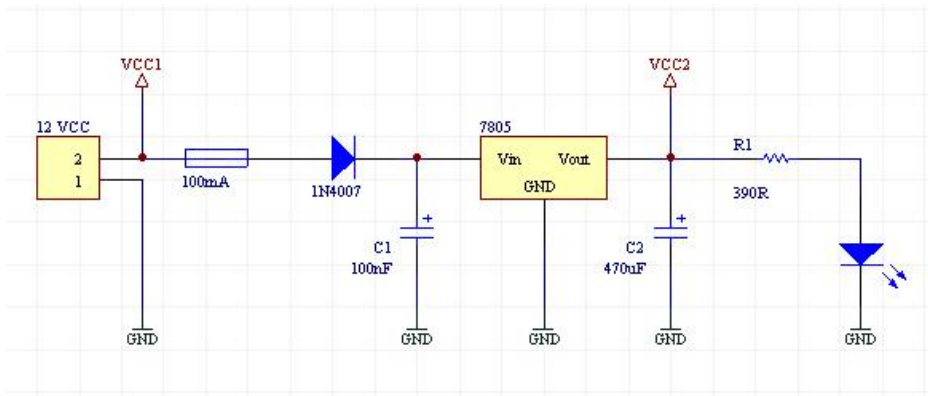


Figura 3.9: Esquema elétrico da fonte de alimentação.

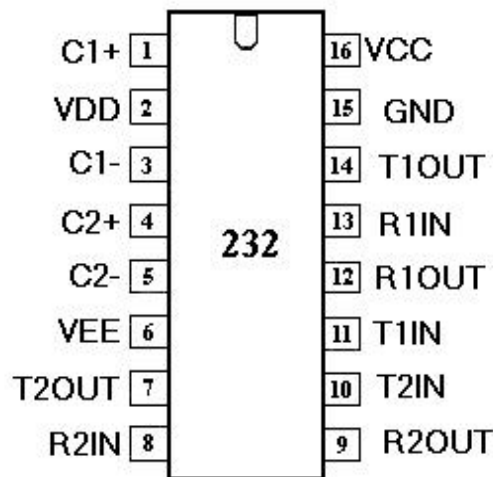


Figura 3.10: Terminais do CI serial RS 232.

CPU, Figura ?? sinalize o funcionamento do programa embutido no microcontrolador. É comum se dizer que o programa está “rodando” quando o LED CPU está piscando. As portas RD0 e RD1 foram configuradas como saída e fornecem os sinais dos pontos 5 e 6 do barramento paralelo dos pontos das celinhas. As portas RD2, RD6 e RD7 não estão sendo utilizadas e estão reservadas para futuras ampliações de celinhas ou outras necessidades de *hardware*. Esta porta D também pode ser utilizada como uma porta paralela de 8 bits. Devido a isto tem-se os termos PSP0 a PSP7, e para se fazer uso desta facilidade utiliza-se também os pinos RE0 (read), RE1 (write) e RE2 (ShipSelect). A facilidade paralela não está sendo utilizada.

Existe ainda outra porta bidirecional de 8 Btis configurável, a porta RE, porém externamente tem-se apenas RE0, RE1 e RE2. RE0 e RE1 não estão sendo utilizadas e estão reservadas para futuras ampliações de celinhas ou outras necessidades de *hardware*. RE2 está configurado como saída e está ligado ao *Buzzer* representado pela Figura ??.

Tabela 3.1: Descrição dos terminais do CI serial RS 232

Pino	Símbolo	Nome e Função
1	$C1+$	Charge Pump
2	VDD	Charge Pump
3	$C1-$	Charge Pump
4	$C2+$	Charge Pump
5	$C2-$	Charge Pump
6	VEE	Charge Pump
7	$T2OUT$	Saída 2 RS 232
8	$R2IN$	Entrada 2 RS 232
9	$R2OUT$	Saída 2 TTL
10	$T2IN$	Entrada 2 TTL
11	$T1IN$	Entrada 1 TTL
12	$R1OUT$	Saída 1 TTL
12	$R1IN$	Entrada 1 RS 232
14	$T1OUT$	Saída 1 RS 232
15	GND	GND
16	VCC	VCC

3.2.4.4 O teclado do Hardware

O equipamento possui um modo de operação independente do microcomputador. Quando é escolhido o modo Transfere no SoftBraille, o texto é gravado na memória da interface, e caso ele seja maior que a quantidade de celinhas disponíveis para a leitura, é necessário um comando para adiantar os próximos caracteres do texto, ou retornar quando o leitor desejar repetir um bloco anteriormente lido, e para isso é utilizado este teclado presente na IBA. Assim, embora estejam previstos quatro botões no teclado, apenas dois foram implementados: um para avançar e outro para retroceder blocos de caracteres de texto em Braille. A Figura ?? representa o circuito elétrico deste módulo do sistema.

Os resistores de 390Ω foram colocados como limitadores de corrente a fim de proteger os terminais do microcontrolador e os resistores de $10k\Omega$, são responsáveis pela estabilidade, evitando que o microcontrolador receba pequenas variações geradas por interferências.

3.2.4.5 O circuito de acionamento

O circuito de acionamento para cada celinha, Figura ??, possui os circuitos integrados ULN 2004 e 74HC373. Neste circuito o CI com 16 pinos, ULN 2004, é um driver de corrente.

Ele funciona da seguinte forma: Cada porta inversora do CI ULN 2004,

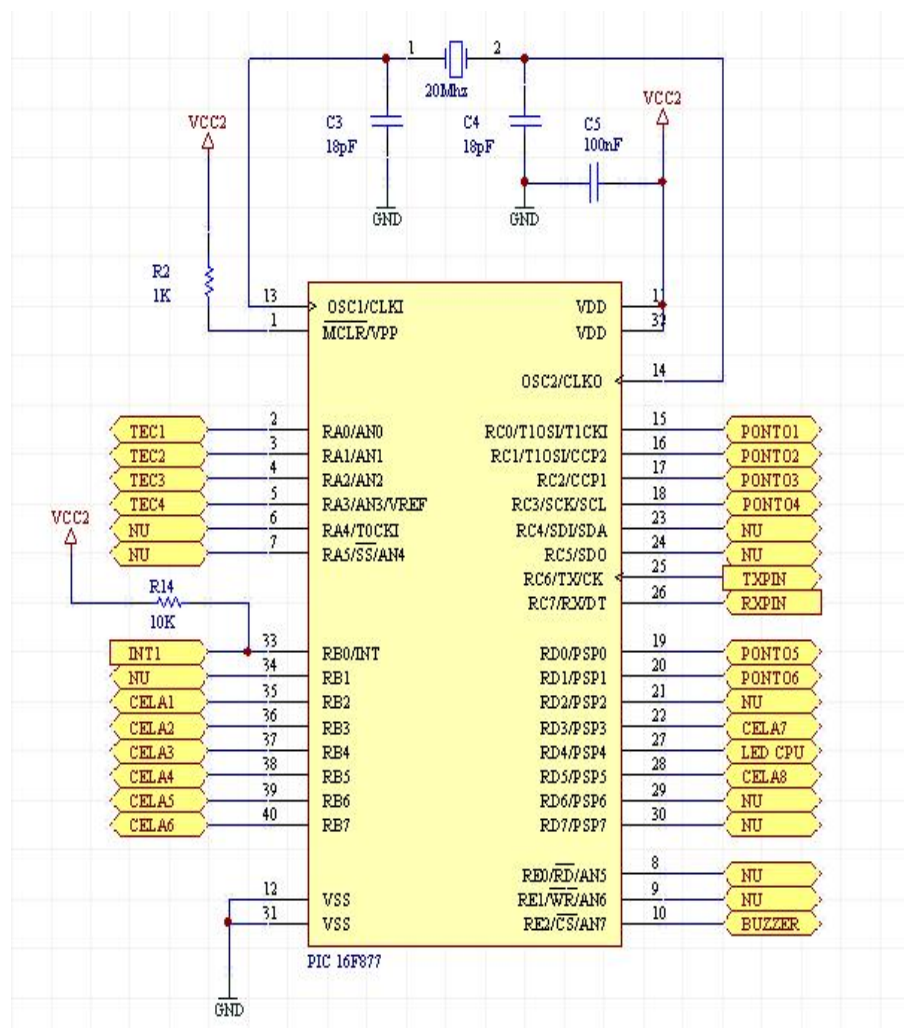


Figura 3.12: Esquema elétrico do módulo de controle da Interface Braille Automatizada.

n , é um nível alto de sinal, ou “*Active High*” como descrito na Tabela ??, que habilita a entrada do CI, e é colocado em “LE” (*latch enable*), no pino 11. Dessa forma, aquilo que houver no barramento, de Ponto 1 a Ponto 6, estará na saída do *Latch*, independente de variações no barramento de Ponto 1 a Ponto 6, e permanecerá assim, até que um novo comando em LE seja colocado pelo microcontrolador. Quando um novo comando chegar em LE, o CI vai “verificar” o que tem no barramento e irá reproduzir o mesmo na saída.

Na Figura ??, percebemos alguns resistores. Aqueles, que estão em série com os LEDs, servem para limitação de corrente afim de evitar danos ao componente. Os outros resistores, em série com as bobinas, têm um papel muito importante neste protótipo. Como as bobinas utilizadas, são de cabeça de impressão de impressoras matriciais, elas consomem muita corrente (cerca de 1A) para fazer com que as agulhas causem o impacto desejado. Assim, como o conjunto mecânico desenvolvido é bastante leve, os resistores foram utilizados para baixar a corrente a um valor de 200mA. Como a dissipação de calor é bastante grande, pelo fato das agulhas não serem utilizadas para impacto, mas para permanecerem acionadas, foram escolhidos resistores de 10W. Estes componentes podem ser vistos na Figura ??, que é a montagem prática do circuito protótipo.

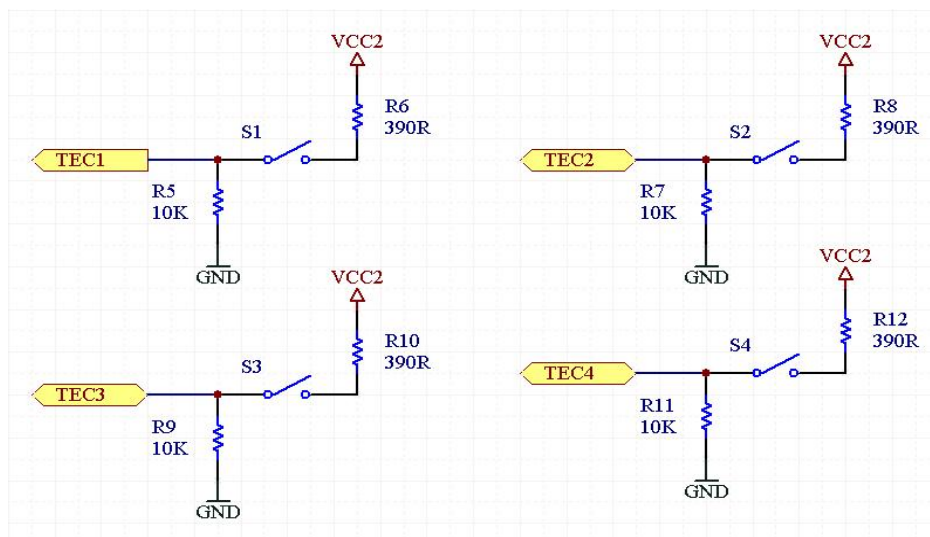


Figura 3.13: Esquema elétrico do teclado.

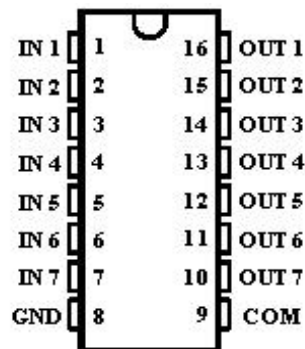


Figura 3.14: Terminais do CI ULN 2004.

3.2.4.6 Circuitos Adicionais

Observando a Figura ?? pode-se notar um circuito contendo um *buzzer*. Este tem a função de fornecer um ruído (Bip) para o usuário. O microcontrolador passa para nível lógico alto no pino 10, sinal *Buzzer*, provocando o chaveamento do transistor BC 548, levando assim o GND ao *Buzzer* provocando o sinal de Bip.

Na figura ?? verifica-se ainda um LED em série com um resistor. Trata-se do LED CPU já comentado anteriormente. Este é o LED que fica piscando enquanto o equipamento está ligado, informando que o circuito está “rodando” o *software* embutido no microcontrolador. Isto é útil para eventuais manutenções no aparelho.

3.2.4.7 A construção do protótipo eletro-mecânico da celinha

Nesse trabalho, embora se tenha um circuito eletrônico para oito celinhas, foi desenvolvido um conjunto eletro-mecânico para duas celinhas apenas para a validação. Esta validação se deu com a ajuda de um aluno DV. Dessa forma, para experimentar

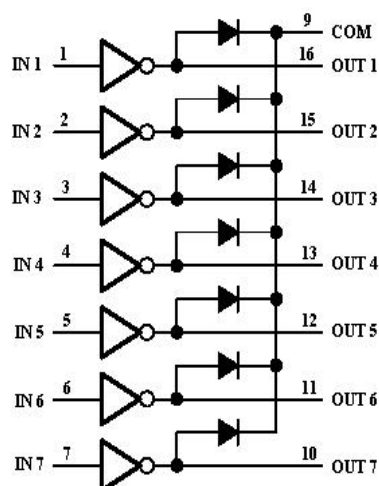


Figura 3.15: Circuito do CI ULN 2004.

a inovação com duas celinhas, foi utilizado o SoftBraille em modo Alfabetiza e assim, utilizando-se apenas as duas celinhas citadas. O processo de validação é abordado no capítulo 4.

Foram utilizados para a construção do protótipo mecânico, cabeças de impressão de impressora matricial, fios de aço do tipo corda de violão, *nylon*, retalhos de alumínio, um reglete como o da Figura ?? e molas diversas.

A superfície plana do protótipo é feita com uma das partes do reglete. Esta é a superfície que proporcionará ao DV o relevo em Braille. Para conseguir fazer a furação, manualmente, foi utilizado o reglete citado, pois do contrário os furos equidistantes a aproximadamente dois milímetros não ficavam certos. Como o reglete possui uma parte com depressões no formato da celinha, estas depressões foram utilizadas como guias.

As cabeças de impressão foram desmontadas e suas agulhas foram retiradas, pois eram curtas. Novas agulhas foram então confeccionadas com fios de aço, do tipo corda de violão. Pedacos de nylon serviram perfeitamente como trava para as molas que forçam as agulhas a retornarem em sua posição original. Outras molas foram utilizadas como guias para que as agulhas não desviem o curso, proporcionando o relevo na superfície.

A estrutura foi feita com pequenos retalhos de alumínio.

Foi então, desta forma, que o protótipo mecânico, tomou a forma da Figura ??, com a placa CPU, Figura ?? presa ao mesmo conjunto, facilitando assim a utilização no momento da validação com os DVs. Pode-se ter uma visão traseira do protótipo na Figura ?. O conjunto das cabeças de impressão, agulhas e molas de guia podem ser vistos na Figura ?.

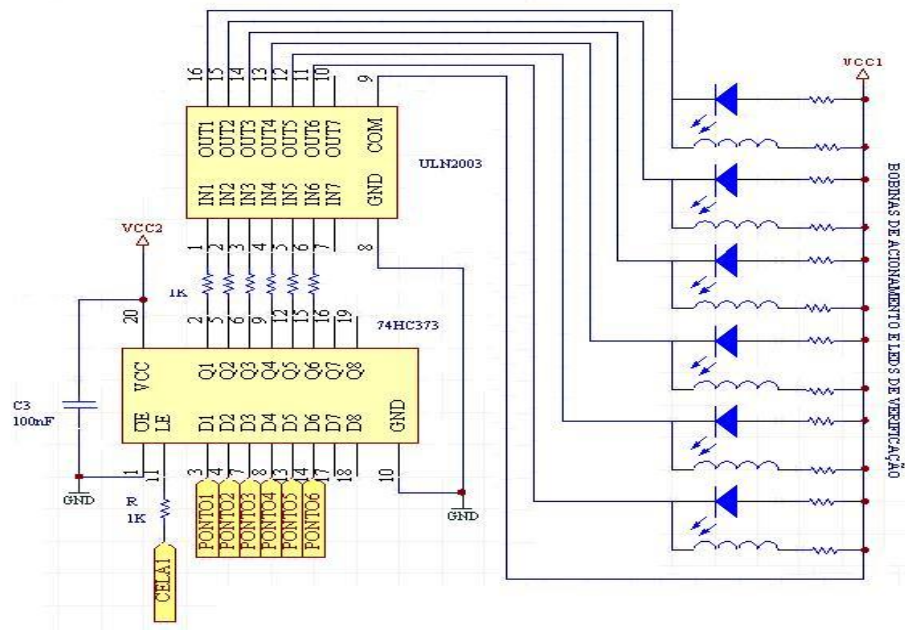


Figura 3.16: Esquema elétrico do circuito de acionamento das bobinas de uma celinha.

3.3 Resumo do capítulo 3

Neste capítulo foi descrita a Interface Braille Automatizada, seus componentes de *Hardware* e *Software* e a montagem do protótipo.

No capítulo seguinte, será visto o trabalho de validação do protótipo desenvolvido, com o auxílio de DVs utilizando o Sistema.

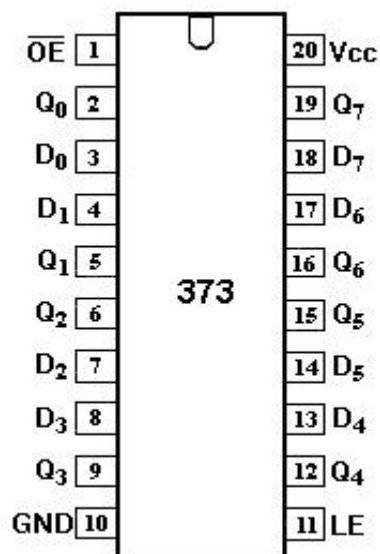


Figura 3.17: Terminais do CI 74HC373.

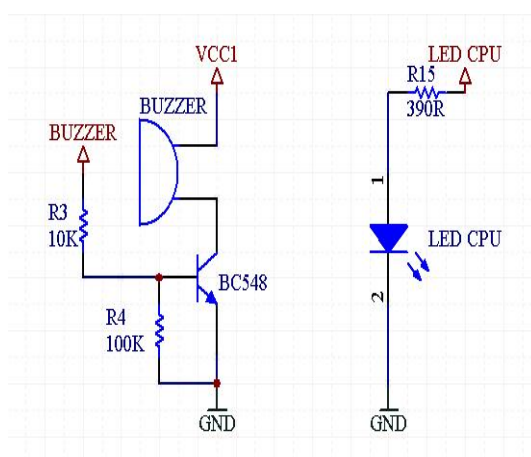


Figura 3.18: Circuito do *buzzer* e LED de verificação do programa.

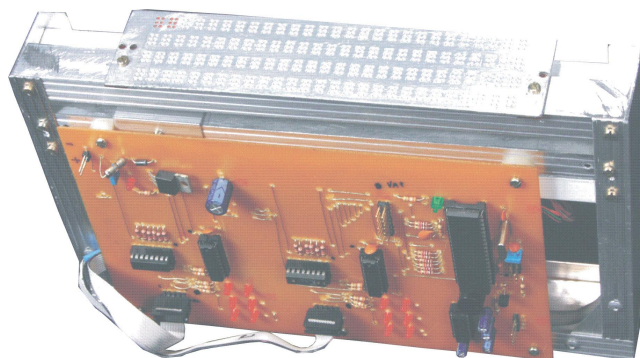


Figura 3.19: Conjunto mecânico da Interface Braille Automatizada.

Tabela 3.2: Descrição dos terminais do microcontrolador 16F877

Pino	Símbolo	Nome e Função
1	\overline{MCLR}/VPP	“Master Clear” Porta de reset (Active HIGH)
2	$RA0/AN0$	TEC1 Porta definida como entrada
3	$RA1/AN1$	TEC2 Porta definida como entrada
4	$RA2/AN2$	TEC3 Porta definida como entrada
5	$RA3/AN3/VREF$	TEC4 Porta definida como entrada
6	$RA4/T0CKI$	NU Porta não utilizada
7	$RA5/\overline{SS}/AN4$	NU Porta não utilizada
8	$RE0/\overline{RD}/AN5$	NU Porta não utilizada
9	$RE1/\overline{WR}/AN6$	NU Porta não utilizada
10	$RE2/\overline{CS}/AN7$	BUZZER Porta definida como saída
11	VDD	Alimentação VCC2
12	VSS	Terra (0V)
13	$OSC1/CLKI$	Oscilador Externo
14	$OSC2/CLKO$	Oscilador Externo
15	$RC0/T1OSI/T1CKI$	Ponto 1 Porta definida como saída
16	$RC1/T1OSI/CCP2$	Ponto 2 Porta definida como saída
17	$RC2/CCP1$	Ponto 3 Porta definida como saída
18	$RC3/SCK/SCL$	Ponto 4 Porta definida como saída
19	$RD0/PSP0$	Ponto 5 Porta definida como saída
20	$RD1/PSP1$	Ponto 6 Porta definida como saída
21	$RD2/PSP2$	NU Porta não utilizada
22	$RD3/PSP3$	Cela 7 Porta definida como saída
23	$RC4/SDI/SDA$	NU Porta não utilizada
24	$RC5/SDO$	NU Porta não utilizada
25	$RC6/TX/CK$	TXPIN Porta de transmissão serial
26	$RC7/RX/DT$	RXPIN Porta de recepção serial
27	$RD4/PSP4$	LED CPU Porta definida como saída
28	$RD5/PSP5$	Cela 8 Porta definida como saída
29	$RD6/PSP6$	NU Porta não utilizada
30	$RD7/PSP7$	NU Porta não utilizada
31	VSS	Terra (0V)
32	VDD	Alimentação VCC2
33	$RB0/INT$	Interrupção externa
34	$RB1$	NU Porta não utilizada
35	$RB2$	Cela 1 Porta definida como saída
36	$RB3$	Cela 2 Porta definida como saída
37	$RB4$	Cela 3 Porta definida como saída
38	$RB5$	Cela 4 Porta definida como saída
39	$RB6$	Cela 5 Porta definida como saída
40	$RB7$	Cela 6 Porta definida como saída

Tabela 3.3: Descrição dos terminais do CI ULN 2004

Pino	Símbolo	Nome e Função
1	$IN1$	Entrada 1 (base)
2	$IN2$	Entrada 2 (base)
3	$IN3$	Entrada 3 (base)
4	$IN4$	Entrada 4 (base)
5	$IN5$	Entrada 5 (base)
6	$IN6$	Entrada 6 (base)
7	$IN7$	Entrada 7 (base)
8	GND	Emissor comum para todas as portas
9	COM	VCC comum para todos os reles
10	$OUT7$	Saída 7 (dreno ou coletor)
11	$OUT6$	Saída 6 (dreno ou coletor)
12	$OUT5$	Saída 5 (dreno ou coletor)
12	$OUT4$	Saída 4 (dreno ou coletor)
14	$OUT3$	Saída 3 (dreno ou coletor)
15	$OUT2$	Saída 2 (dreno ou coletor)
16	$OUT1$	Saída 1 (dreno ou coletor)

Tabela 3.4: Descrição dos terminais do CI 74HC373

Pino	Símbolo	Nome e Função
1	\overline{OE}	Habilita a Saída 3-state receber a Entrada (Active Low)
2, 5, 6, 9, 12, 15, 16, 19	Q_0 a Q_7	Saídas 3-state Latch
3, 4, 7, 8, 13, 14, 17, 18	D_0 a D_7	Entrada de Dados
10	GND	Terra (0V)
11	LE	Habilita latch input (Active HIGH)
20	V_{CC}	Alimentação positiva da fonte

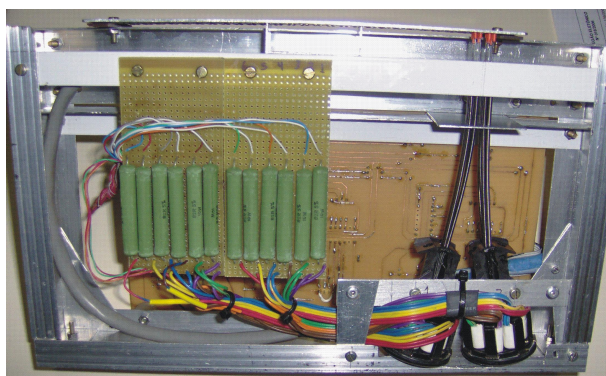


Figura 3.20: Vista traseira da IBA.

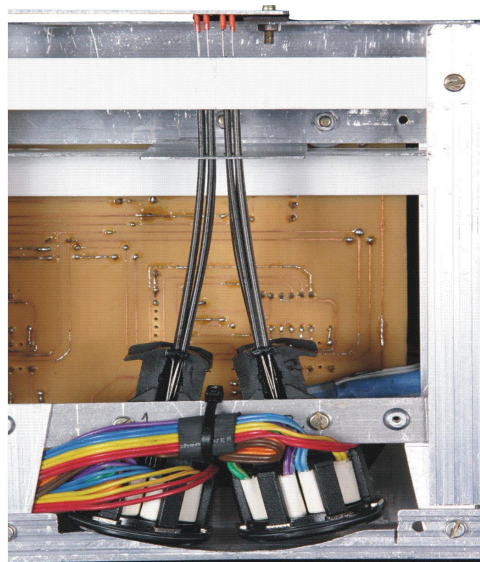


Figura 3.21: Vista das cabeças, agulhas e molas de guia.

4 Validação

Para a validação, foi preciso contar com profissionais da área de educação e também de alunos DVs. Grande foi o apoio do CAEDV - Centro de Atendimento Especializado para DVs - da Escola Municipal Borges de Medeiros, Educação Infantil e Ensino Fundamental do Município de Toledo - PR.

4.1 Teste da Interface Braille Automatizada por um DV

O aluno Deficiente Visual, L. T., Figura ??, utilizou a interface e alguns professores acompanharam. Suas reações, comentários e sugestões serão descritos nos tópicos seguintes.

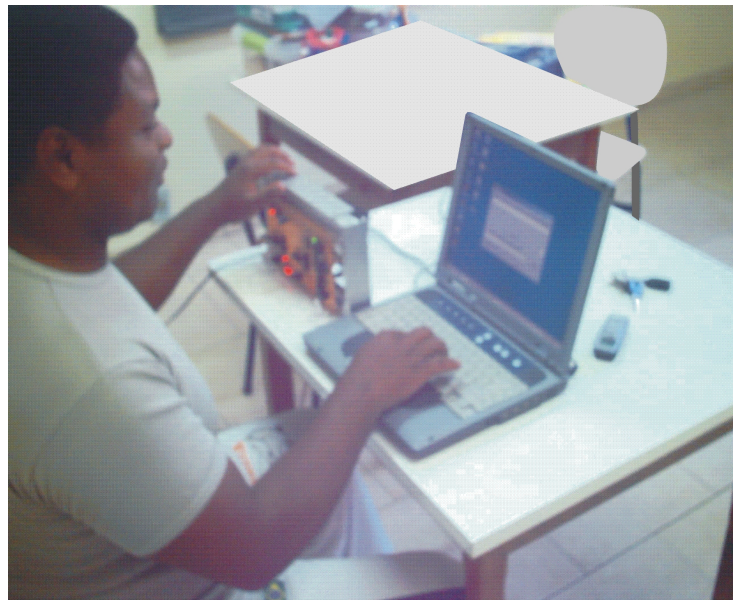


Figura 4.1: Aluno DV utilizando a Interface Braille Automatizada.

4.1.1 O início

Ao executar o programa, o aluno ouviu a mensagem de saudação e a ajuda do sistema. Imediatamente comparou com o áudio do sistema DoxVox ^[?] e elogiou. Em sua opinião, o áudio é bom, sendo de qualidade superior. O uso do sistema DoxVox é muito comum entre os alunos DVs, porém é um som sintetizado, por isso há uma diferença com o áudio desse projeto.

Como o interesse é verificar a automatização da celinha do alfabeto Braille e a Interface Braille Automatizada, e o protótipo montado na prática possui apenas duas celinhas, foi escolhido o modo alfabetiza para o teste prático com o DV.

4.1.2 A sensação tátil

Com o modo alfabetiza em funcionamento, o DV começou a utilizar e a entender seu funcionamento. Assim, a cada tecla digitada ele leu, por meio do tato, os caracteres das celinhas da Interface Braille Automatizada, não apresentando dificuldades no procedimento.

Segundo o relato do DV, a leitura dos caracteres nas celinhas automatizadas obteve resultado positivo.

4.1.3 Comparação entre IBA e impressão Braille em papel pelo DV

Para o aluno, não existiu diferença significativa, e ressaltou que nada há nas celinhas automatizadas que fique aquém da qualidade obtida com o braille impresso no papel em auto-relevo.

4.1.4 O manuseio do SoftBraille pelo DV

O DV que auxiliou no trabalho de validação possui um microcomputador e está familiarizado com o teclado. Ele utilizou o SoftBraille, sem muita ajuda dos videntes à sua volta, sentindo certa diferença entre o teclado do *notebook*, utilizado na ocasião, e o teclado que ele utiliza em sua casa. Em sua opinião, e dos professores presentes, o módulo alfabetiza é muito interessante e possível de ser utilizado como ferramenta de auxílio ao aprendizado. Um dos professores, fez uma comparação entre o sistema, no modo alfabetiza, e um papel com alguns recortes colados para dar o alto-relevo, que também serve para ensinar o Braille, e concluiu que este trabalho é importante na inclusão digital do DV e é uma ferramenta muito útil e ainda melhor que certos recursos utilizados até o momento.

4.1.5 Sugestões durante a utilização

O aluno sugeriu que a IBA ficasse disposta como um reglete no próprio teclado do microcomputador, logo acima das teclas de função.

Uma das professoras presentes, sugeriu que a opção de ajuda possa ter uma forma de controle de velocidade para que os alunos que tenham menos experiência com o microcomputador possam entender mais claramente as informações.

4.2 Conclusão da validação

Embora tenha sido descrito o teste do protótipo da interface por apenas um DV, e alguns professores especializados do CAEDV, o projeto participou do I Seminário de Inserção Social - Educação e Trabalho das pessoas com Deficiência Visual, em 06/10/2006, e foi experimentado por dezenas de pessoas, entre as quais DVs e videntes professores especializados em educação inclusiva e familiares de DVs, conforme declaração no Anexo B.

Durante o evento, diversas perguntas foram feitas aos usuários que utilizaram o sistema. Esta pesquisa pode ser encontrada no Anexo D.

Concluiu-se que o trabalho atendeu as expectativas, embora se tenha pela frente um grande trabalho de pesquisas e desenvolvimento para o aprimoramento do sistema.

4.3 Resumo do Capítulo 4

Neste capítulo foi descrito o processo de validação do trabalho desenvolvido, com o auxílio de DVs e profissionais de ensino utilizando o Sistema.

No capítulo seguinte serão abordados o estado atual e trabalhos futuros do objeto deste trabalho tanto em *Software*, quanto em *Hardware*.

5 *Estado atual e trabalhos futuros*

5.1 Inovação tecnológica

O trabalho apresentou-se novo, com muitas reações de surpresa em todos os ambientes apresentados, principalmente durante o rápido processo de validação.

5.1.1 Depósito de Patente

O mentor deste projeto é o Prof. Dr. Walter Germanovix, que aceitou ser orientador desta dissertação de mestrado. O Prof. Walter me convidou para participar como co-autor no depósito de patente, uma vez que este trabalho está ligado ao desenvolvimento do *Software e Hardware* para validar o sistema proposto.

O Depósito de Patente foi protocolado conforme diretrizes indicada pela Intuel - Incubadora Internacional de Empresas de Base tecnológicas da UEL, sob número PI0403017-6, em 07/07/2004.

5.2 Perspectivas de mercado

O mercado se mostra muito promissor, pois muitos equipamentos para DVs são importados e caros. Segundo relatos dos profissionais de ensino especializados em educação especial, uma impressora Braille, modelo JuLiet, pode custar aproximadamente R\$ 26.000,00. De acordo com estes profissionais, muitas escolas necessitam de ferramentas para trabalhar com DVs, e é direito deles que as entidades de ensino estejam munidas de professores que os atendam e que tenham aparelhos que lhes auxiliem o aprendizado. Este projeto, pode ser uma excelente ferramenta caso as escolas e universidades pudessem contar com este recurso. O custo desta interface não foi estimado, pois o acionamento dos pinos, do protótipo, foi desenvolvido com bobinas de impressoras matriciais comuns, não sendo portanto um modelo comercial, porém com estimativa de valor bem aquém do equipamento citado.

5.3 Futuras contribuições ao trabalho

Outros tipos de acionamentos, das agulhas das celinhas, poderiam ser desenvolvidos com o intuito de poder aprimorar e facilitar a leitura dos caracteres pelo tato do DV. Por exemplo, o núcleo poderia subir girando para melhorar a sensibilidade do tato do DV. Superfícies rugosas também podem auxiliar na sensibilidade. Outras possibilidades de alto-relevo poderiam ser utilizadas, como por exemplo a utilização de deformações piezoelétricas.

Bobinas com dimensões menores deverão ser desenvolvidas, para tornar o aparelho portátil. Isto também poderia fazer com que o sistema consuma menos corrente elétrica. Hoje, devido ao tipo de bobinas utilizadas, o sistema consome aproximadamente 2A com apenas duas celinhas.

A Interface Braille Automatizada poderá ficar disposta como um reglete no próprio teclado do microcomputador, logo acima das teclas de função.

Novos estudos, com *softwares* por linhas de comando e não por janelas, pode proporcionar o desenvolvimento de um *notebook* para DVs. O *software* livre Linux pode ser indicado para isso. No lugar da tela, poderiam ser diversas linhas e colunas de pinos que fossem capazes de, em relevo, representar inclusive figuras.

Desenvolvimento do SoftBraille em outras linguagens de programação que possam ser utilizados por outros sistemas operacionais, como o Linux.

Melhorias no SoftBraille que venham a permitir que o DV acesse grupos de bate-papo na internet.

Conjunto de lâmpadas acionadas por reles, no lugar do conjunto de LEDs, pode ser utilizado para trabalhos com pessoas de baixa visão ou de visão sub-normal. Neste caso, não se utiliza a celinha automatizada, mantendo-se os demais *softwares* e circuitos.

A utilização de técnicas de síntese de voz tornará o sistema mais completo e aplicável a outras necessidades especiais.

5.4 Resumo do Capítulo 5

Neste capítulo, foi feita uma descrição do estado atual e perspectivas futuras do objeto deste trabalho tanto em *Software* quanto em *Hardware*.

6 *Conclusão*

Os sistemas computacionais têm seu espaço garantido na sociedade moderna, sendo não somente uma ferramenta, mas uma necessidade. Assim, diante desse avanço tecnológico, o Deficiente Visual não pode ser excluído e condenado a não ter acesso a esta tecnologia. O uso do computador pelo Deficiente Visual é um direito que lhe proporciona a inclusão digital.

A ferramenta de leitura/escrita mais utilizada pelos Deficientes Visuais, é o Alfabeto Braille. Contudo o espaço tomado por livros impressos em Braille é muito grande e o custo de impressão é alto. Portanto, um sistema que possa automatizar os caracteres Braille pode proporcionar inúmeras vantagens.

O conjunto de *hardware* e *software*, desenvolvidos no Laboratório de Engenharia Biomédica do Departamento de Engenharia Elétrica da UEL, mostrou ser eficiente e indicado para auxiliar os Deficientes Visuais, o que mostra que deve haver um esforço de toda a sociedade para que a tecnologia seja empregada. Isso foi comprovado pelos testes efetuados com os usuários Deficientes Visuais e professores da área que testaram o sistema.

Muito deve ser aperfeiçoado, como torná-lo mais anatômico ao uso para ter mais praticidade. Novas técnicas para o conjunto mecânico podem reduzir o tamanho e melhorar a portabilidade. Outras ferramentas de programação podem ser utilizadas para adequar o conjunto a outros sistemas operacionais. Sistemas de síntese de voz podem ajudar a tornar o sistema mais amigável ao Deficiente Visual e útil para muitas outras aplicações.

Referências Bibliográficas

- 1 ALMEIDA, Maria da Glória de S. **Prontidão para alfabetização através do sistema braille**. Rio de Janeiro: Instituto Benjamin Constant, 1995.
- 2 BARRAGA, N. **Baja Vision: Evaluacion diagnostica, Procedimiento de Valoracion Diagnostica, Guia del Maestro para el Desarrollo de la Capacidad de Aprendizaje Visual y la Utilizacion de la Pobre Vision, dor.** Programa para desarrollar la eficiencia en el funcionamiento visual. Madrid: ONCE, 1986.
- 3 BARRAGA, N. **Disminuidos Visuales y Aprendizaje: Enfoque Evolutivo**. Madrid: ONCE, 1975.
- 4 BARRERA, Alberto Rodríguez. **Computadoras contra la ceguera**. Lunes, disponível em: <<http://www.elmundo.com.ve/ediciones/2002/10/14/p1-15s1.htm>> acesso em 25/08/2004.
- 5 BRUNO, Marilda Moraes Gracia. **O deficiente visual na classe comum**. São Paulo: Secretaria Estadual da Educação, 1987.
- 6 CARVALHO, José Oscar Fontanini de. **Interfaces para o deficiente visual**. Campinas, PUC-Revista informédica, v. 1 p. 5-11, disponível em: <<http://www.epub.org.br/informed/defic.htm>> acesso em 26/08/2005.
- 7 CHAPMAN, E.; STONE, J. **The Visually Handicapped Child in Your Classroom: Special Needs in Ordinary Schools**. London: Cassell Educational, 1988.
- 8 COLL, César Palacios J. **Necessidades educativas especiais e aprendizagem escolar**. Porto Alegre: Artes Médicas, 1995.
- 9 CONDE, Antônio João Menescal. **A pessoa portadora de deficiência visual: seu movimento e seu mundo**. Rio de Janeiro: IBC, 1992.
- 10 CUTSFORTH, T. **O cego na escola e na sociedade: Campanha Nacional de Educação de Cegos**. Brasília: MEC, 1969.
- 11 FEDF. **Atendimento educacional ao aluno portador de deficiência da visão**. Brasília: Secretaria de Educação do Distrito Federal, 1994.
- 12 FICHTNER, D. **Como criar un niño ciego: Guia para padres de niños ciegos y para quienes trabajan con niños de edad pre-escolar**. Comité Regional Latino-Americano do ICEVH, 1979.
- 13 FRAIBERG, S. **Insight from the blind Library of Congress**. Washington: 1972.

- 14 GARCIA A. **La Vision Subnormal:** Una Nueva Especializacion. in Gaceta Optica nº 225. Comissão de Braille. Lisboa, 2000.
- 15 GIL, Marta. **Cadernos da TV Escola:** Deficiencia Visual. Brasilia: MEC - Secretaria de Educação a Distância, 2000.
- 16 HALLIDAY, Carol. **Crescimento, aprendizagem e desenvolvimento da criança visualmente incapacitada do nascimento à idade escolar.** São Paulo: Fundação para o Livro do Cego no Brasil, 1975.
- 17 INSTITUTO BENJAMIN CONSTANT. **Guia teórico para alfabetização em braille.** Rio de Janeiro: IBCENTRO, 1995.
- 18 INSTITUTO BENJAMIN CONSTANT/MEC. **Revista Benjamin Constant.** v. 2, n. 3. Rio de Janeiro: IBCENTRO, 1996.
- 19 INSTITUTO BENJAMIN CONSTANT/MEC. **Revista Benjamin Constant.**v. 3, n. 6. Rio de Janeiro: IBCENTRO, 1997.
- 20 INSTITUTO BENJAMIN CONSTANT/MEC. **Revista Benjamin Constant.** v. 3, n. 7. Rio de Janeiro: IBCENTRO, 1997.
- 21 INSTITUTO BENJAMIN CONSTANT/MEC. **Revista Benjamin Constant.** v. 4, n. 9. Rio de Janeiro: IBCENTRO, 1998.
- 22 JAEKLE, R. **Orientation and Mobility:** Proceedings of the 6th Quinquennial Conference of the International Council for Education of the Visually Handicapped. Paris: ICEVH, 1977.
- 23 JENSEN, L.R.; BORGES, J.A. **Cegos e Computador:** Uma Interação que Explora o Potencial do Desenho. Anais do Seminário de Engenharia da UFF, 1999.
- 24 KIRK, S. **Educating exceptional children.** 2. ed. Houghton Hifflin Company Poston, 1972.
- 25 LOPEZ, Ricardo Aldabó. **Sistemas de Redes para Controle e Automação.** Rio de Janeiro: Book Express, 2000.
- 26 LOWENFILE, B. **Our blind children.** Illinois: Charles Thomas Publishers, 1964.
- 27 MANTOAN, Maria Teresa Eglér. **A integração de pessoas com deficiência:** contribuições para uma reflexão para o tema. São Paulo: SENAC, 1997.
- 28 MASINI, Elsie S. **O perceber e o relacionar-se do deficiente visual.** Brasília: Ministério da Justiça, 1994.
- 29 MAZOTTA, Marcos José da Silveira. **Trabalho docente e formação de professores de educação especial.** São Paulo: EPU, 1993.
- 30 MAZOTTA, Marcos José da Silveira. **Educação especial no Brasil:** história e políticas públicas. São Paulo: Cortez, 2001.
- 31 MEC. **A cegueira trocado em miúdos.** Brasília: Fundação de Assistência ao Educando, 1988.

- 32 MEC. **A educação especial no Brasil**. Brasília: Secretaria de Educação Especial, 1994.
- 33 MEC. **Projeto CAP**. MEC, disponível em: <<http://www.mec.gov.br/seesp/apoio.shtm>> acesso em 25/08/2005.
- 34 MEC, CENESP-IBC. **Proposta curricular para deficientes visuais**. Pré-escolar. Rio de Janeiro: MEC, 1982.
- 35 MEC. **Revista Integração**. v. 23. Brasília: SEESP, 2001.
- 36 MICROCHIP. **Product Line Card: Including Development Tools**. Microchip Technology Inc, 2001.
- 37 MICROSOFT. **Visual Basic: Language Reference**. Microsoft Corporation, 1993.
- 38 MICROSOFT. **Visual Basic: Programmer's Guide**. Microsoft Corporation, 1993.
- 39 MINISTERIO DE EDUCACIÓN Y CIENCIA. **Alumnos com necesidades educativas especiales y adaptaciones curriculares**. España, Madrid: 1992.
- 40 MELO, José Marques de; SATHLER, Luciano. **Direitos à Comunicação na Sociedade da Informação**. São Bernardo do Campo: UMESP, 2005.
- 41 MENDONÇA, A.; REINO, Vítor. **A Leitura do Braille: O Sistema Braille aplicado à Língua Portuguesa**. Lisboa: ACAPO, 1994.
- 42 MORRIS, M. **Blindness in children**. Chicago: University of Chicago Press, 1975.
- 43 NELLY, Garcia. **Educação Especial FEUSP**. Universidade de São Paulo, disponível em: <<http://www.braillevirtual.fe.usp.br/>> acesso em 24/08/2005.
- 44 NETO, João Lucas. **A deficiência visual e você**. Porto Alegre: 1988.
- 45 NOVI, Rosa Maria. **Exemplos de vida**. Londrina: Grafman, 1999.
- 46 PARKER, Steve. **Conviver com a cegueira**. São Paulo: Scipione, 1994.
- 47 PORTO, B. C. **Webvox/Intervox: tese de mestrado em informática**. Instituto de Matemática e NCE/UFRJ - Cap. 3 - março/2001.
- 48 REINO, Vítor. **Ensino/Aprendizagem do Braille: Colóquio "O Braille que temos, o Braille que queremos"**. Lisboa: Comissão de Braille, 2000.
- 49 SECRETARIA DE ESTADO DA EDUCAÇÃO. **Diretrizes Teórico- Metodológicas para a Educação Especial: Documento Preliminar, Área Visual**. Curitiba: Departamento de Educação Especial, 1994.
- 50 SECRETARIA DE ESTADO DA EDUCAÇÃO. **Fundamentos Teórico - Metodológicos para a Educação Especial**. Curitiba: Departamento de Educação Especial, 1994.
- 51 SENE. **Atividades físicas adaptadas ao deficiente visual**. Revista Integração. Brasília, v. 3, n. 7, 1991.

- 52 SILVA, António; SERRA, Daniel. **LERPARAVER**: O portal da visão diferente. Lerparaver, disponível em: <<http://www.lerparaver.com/index.html>> acesso em 29/08/2005.
- 53 SOUZA, David José de. **Desbravando o PIC**: ampliado e atualizado para PIC 16F628A. São Paulo: Érica, 2003.
- 54 SOUZA, David José de. **Conectando o PIC**: explorando recursos avançados. São Paulo: J. J. Carol Editora, 2002.
- 55 TORRES, Gabriel. **HARDWARE**: Curso Completo. Rio de Janeiro: Axcel Books, 2001.
- 56 UNIVERSIDADE FEDERAL DO RIO DE JANEIRO. **Projeto DOSVOX**. Núcleo de computação eletrônica, disponível em: <<http://intervox.nce.ufrj.br/dosvox/>> acesso em 25/08/2004.
- 57 WERNECK, Cláudia. **Ninguém mais vai ser bonzinho na sociedade inclusiva**. Rio de Janeiro: WVA, 1997.
- 58 Wikipedia. **A Enciclopédia livre**. Wikipedia, disponível em: <<http://gl.wikipedia.org/wiki/Braille>> acesso em 26/08/2005.

APÊNDICE A – Rotinas principais do Visual Basic

Muitas rotinas são repetitivas e não serão apresentadas por inteiro. O que é válido para um caractere é válido para todos os outros. Portanto, mostrarei aqui somente o corpo principal do software.

O programa foi dividido em dois formulários e três módulos mostrados a seguir.

A.1 Formulário principal (FrmPrincipal)

Option Explicit

```
Private Sub Command1_Click()
Dim i As String Dim Cont As Integer
Dim cont1 As Integer
Dim word As New word.Application
Dim TextLine As String

If ActiveControl.Name = "Text2" Then
    Text2.SetFocus
    Exit Sub
End If

If Trim(Text1.Text) = "" Then
    PlaySound CaminhoSons & "not_find.WAV"
    MsgBox "O caminho não existe!", 48, "Tente novamente"
Else
    If Trim(Text1.Text) > "" Then
        extensao = Right(Text1.Text, 4)
        i = Right(Text1.Text, 4)
        cont1 = 4
        Do Until i = "/"
```

```
        On Error GoTo fim
        Cont = 1
        cont1 = cont1 + Cont
        nome = Right(Text1.Text, cont1)
        i = Right(Text1.Text, cont1)
        i = Left(i, 1)
    Loop
    Caminho = Left(Text1.Text, (Len(Text1.Text) - (cont1 - 1)))
    cont1 = cont1 - 1
    nome = Right(nome, cont1)
    Cont = Len(nome)
    cont1 = cont1 - 4
    nome = Left(nome, cont1)
    cont1 = 4
    Cont = 0
    cont1 = 0
    If FileExist(CaminhoTexto & nome & ".txt") = True Then
        PlaySound CaminhoSons & "existente.WAV"
        FrmMsgBox.Show
        Exit Sub
    End If
    If extensao = ".doc" Or extensao = ".rtf"
    Or extensao = ".xls" Or extensao = ".htm" Then
        If ConverteWordDocumento(Text1.Text, wdFormatText,
            CaminhoTexto & nome & "1.txt") Then
            If RemoveEspacos(CaminhoTexto & nome
                & "1.txt", nome) Then
                PlaySound CaminhoSons & "converte_ok.WAV"
                MsgBox " Arquivo convertido com sucesso ! "
                Text1.Text = ""
            End If
            Else
                PlaySound CaminhoSons & "not_find.WAV"
                MsgBox "O caminho não existe!", 48,
                    "Tente novamente"
                Text1.Text = ""
            End If
        End If
    End If
End If

Text1.SetFocus
Exit Sub
```

fm:

```
PlaySound CaminhoSons & "not_find.WAV"  
MsgBox "O caminho não existe!", 48, "Tente novamente"  
Text1.Text = ""  
cont1 = 4  
Cont = 0  
cont1 = 0  
Text1.SetFocus  
Exit Sub  
End Sub
```

```
Private Sub Command2_Click()
```

```
MSComm1.Output = Chr(255)  
PlaySound CaminhoSons & "dindon.WAV"  
    If MSComm1.PortOpen = True Then  
        MSComm1.PortOpen = False  
    End If  
End  
End Sub
```

```
Private Sub Command3_Click()
```

```
Command1.Visible = False  
Command3.Visible = False  
Command6.Visible = False  
Command7.Visible = False  
Command8.Visible = False
```

```
PlaySound CaminhoSons & "alfabetiza.WAV"
```

```
Text1.Text = ""  
X = True  
Text1.SetFocus  
End Sub
```

```
Private Sub Command5_Click()
```

```
PlaySound CaminhoSons & "help.WAV"  
Text1.SetFocus  
End Sub
```

```
Private Sub Command6_Click()
MSComm1.Output = Chr(250)
MSComm1.Output = Chr(8)

Command1.Visible = False
Command3.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
PlaySound CaminhoSons & "leitura.WAV"
If Val(Text2.Text) = 0 Then
    Command7_Click
    Exit Sub
End If
If (FileExist(CaminhoTexto & Text1.Text & ".txt") = True) Then

    Ind = Text2.Text
    Aux = 1
    Tam = 0
    X = 0

    Open CaminhoTexto & Text1.Text & ".txt" For Input As #1

    Do While Not EOF(1)
    Input #1, Linha
    Aux = 1
    Do While Aux < Len(Linha)
        If Aux <= Len(Linha) Then
            Var = Mid((Linha), Aux, Ind)
            Aux = Aux + Ind
            Tam = Tam + 1
            Var = ""
        End If
    Loop
    Loop
Close #1
```

```
Ind = Text2.Text
Aux = 1
ReDim Matriz(Tam)
X = 0

Open CaminhoTexto & Text1.Text & ".txt" For Input As #1
Do While Not EOF(1)
    Input #1, Linha
    Aux = 1
    Do While Aux < Len(Linha)
        If Aux <= Len(Linha) Then
            If Trim(Mid((Linha), Aux, Ind)) <> "" Then
                Matriz(X) = Mid((Linha), Aux, Ind)
            End If
            Aux = Aux + Ind
            X = X + 1
        End If
    Loop
Loop
Close #1
List1.Clear

For Ind = 0 To Tam
    List1.AddItem Matriz(Ind)
Next Ind
List1.SetFocus

Else
    PlaySound CaminhoSons & "not_find.WAV"
    MsgBox "O caminho não existe!", 48, "Tente novamente"
    Text1.Text = ""
    Command1.Visible = True
    Command3.Visible = True
    Command6.Visible = True
    Command7.Visible = True
    Command8.Visible = True
    Text1.SetFocus
End If
End Sub

Private Sub Command7_Click()
    bot_cel = True
```

```
        PlaySound CaminhoSons & "QuantCel.WAV"
        Text2.Enabled = True
        Command1.Enabled = False
        anterior = Text2
        q = 1
        Text2 = ""
        Text1 = ""
        Text2.SetFocus
End Sub

Private Sub Command8_Click()
    Dim auxiliar As Integer
    Dim Trans As String
    Dim Cont As Integer

    Cont = 0
    auxiliar = 1
    If FileExist(CaminhoTexto & Text1.Text & ".txt") = True Then
        Open CaminhoTexto & Text1.Text & ".txt" For Input As #1
        Do While Not EOF(1)
            If Cont <= 240 Then
                Trans = Input(1, #1)
                Cont = Cont + 1
                MSComm1.Output = Trans
            Else
                PlaySound CaminhoSons & "grande.WAV"
                MsgBox "A quantidade de caracteres passou
do limite permitido, 240!!", 48, "Atenção"
                Text1.Text = ""
                Text1.SetFocus
                Close #1
                Exit Sub
            End If
        End Do
    End If

    Loop
    Close #1

Else
    PlaySound CaminhoSons & "not_find.WAV"
    MsgBox "O caminho não existe!", 48, "Tente novamente"
    Text1.Text = ""
    Text1.SetFocus
End If
```

```
        auxiliar = 2
End If

Text1.Text = ""
Text1.SetFocus
If auxiliar <> 2 Then
    PlaySound CaminhoSons & "trans.WAV"
End If
auxiliar = 1
End Sub

Private Sub Form_Load()
    Dim n$ Dim count As Integer
    Dim msg As String
    Dim celinhas As Integer

    Top = 300 Height = 4995 Width = 7380 Left = (Screen.Width - Width) % / 2

    Text2.Enabled = False

    bot_cel = False

    q = 0

    Open App.Path & "celinha.txt" For Input As #1
    If Not EOF(1) Then
        Input #1, celinhas
    End If
    Close #1
    If Val(celinhas) <> 0 Then
        Text2.Text = celinhas
    Else
        Text2.Text = ""
    End If

    QuantCel = Text2.Text
    If MSComm1.PortOpen <> True Then
        MSComm1.CommPort = 1
        MSComm1.Settings = "2400,n,8,2"
```



```
        MSComm1.PortOpen = True
        MSComm1.RThreshold = 1
        MSComm1.InputLen = 2

        PortaSerial
    End If

    PlaySound CaminhoSons & "welcome.WAV"
End Sub

Private Sub List1_KeyDown(KeyCode As Integer, Shift As Integer)
    Dim Envia As String
    If KeyCode = 40 Then
        Ind = (List1.ListIndex) + 1
        If Val(Ind) <= 0 Then
            Envia = List1.List(Ind)
            MSComm1.Output = Envia
        End If
    End If
End Sub
If KeyCode = 38 Then

    If Val(Ind) >= 0 Then
        Ind = List1.ListIndex - 1
        Envia = List1.List(Ind)
        MSComm1.Output = Envia
    End If End If

End Sub

Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
    Label2.Caption = KeyCode
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer) Label1.Caption = KeyAscii
If Label1.Caption = "48" Then
    letra = "0"
    PlaySound CaminhoSons & "0.WAV"
End If
```

```
If Label1.Caption = "49" Then
    letra = "1"
    PlaySound CaminhoSons & "1.WAV"
End If

If Label1.Caption = "50" Then
    letra = "2"
    PlaySound CaminhoSons & "2.WAV"
End If

If Label1.Caption = "64" Then
    letra = ""
    PlaySound CaminhoSons & "arroba.WAV"

If Label1.Caption = "35" Then
    letra = "#"
    PlaySound CaminhoSons & "number.WAV"
End If

If Label1.Caption = "36" Then
    letra = "$"
    PlaySound CaminhoSons & "cifrao.WAV"
End If

If Label1.Caption = "91" Then
    letra = "["
    PlaySound CaminhoSons & "abre_colchetes.WAV"
End If

If Label1.Caption = "93" Then
    letra = "]"
    PlaySound CaminhoSons & "fecha_colchetes.WAV"
End If

If Label1.Caption = "124" Then
    letra = "_"
    PlaySound CaminhoSons & "barra_vertical.WAV"
End If
```

```
If Label1.Caption = "224" Then
    letra = "à"
    PlaySound CaminhoSons & "acrase.WAV"
End If
```

```
If Label1.Caption = "225" Then
    letra = "á"
    PlaySound CaminhoSons & "aagudo.WAV"
End If
```

```
If Label1.Caption = "226" Then
    letra = "â"
    PlaySound CaminhoSons & "acircunflexo.WAV"
End If
```

```
If Label1.Caption = "227" Then
    letra = "ã"
    PlaySound CaminhoSons & "atil.WAV"
End If
```

```
If Label1.Caption = "97" Then
    letra = "a"
    PlaySound CaminhoSons & "a.WAV"
End If
```

```
If Label1.Caption = "98" Then
    letra = "b"
    PlaySound CaminhoSons & "b.WAV"
End If
```

```
If Label1.Caption = "99" Then
    letra = "c"
    PlaySound CaminhoSons & "c.WAV"
End If
```

```
If Label1.Caption = "231" Then
    letra = "ç"
    PlaySound CaminhoSons & "cedilha.WAV"
End If
```

```
If Label1.Caption = "100" Then
    letra = "d"
    PlaySound CaminhoSons & "d.WAV"
End If
```

A.2 Formulário de mensagens (Frm_Msg_Box)

```
Private Sub CmdCancelar_Click()
    PlaySound CaminhoSons & "cancelado.WAV"
    FrmPrincipal.Text1.Text = ""
    FrmPrincipal.Text1.SetFocus
    Unload Me
End Sub
```

```
Private Sub CmdRenomear_Click()
    Label2.Visible = True
    TxtNovoNome.Visible = True
    PlaySound CaminhoSons & "novoarq.WAV"
    'MsgBox "Digite o novo nome do arquivo e pressione enter!", 48, "Atencao"
    TxtNovoNome.SetFocus
    'Unload Me
End Sub
```

```
Private Sub CmdSob_Click() Dim NomeArq As String
    Kill (CaminhoTexto & nome & ".txt") 'apaga o aquivo antigo
    Unload Me
    If extensao = ".doc" Or extensao = ".rtf"
    Or extensao
    = ".xls" Or extensao = ".htm" Then
        If ConverteWordDocumento(FrmPrincipal.Text1.Text,
        wdFormatText, CaminhoTexto & nome & "1.txt") Then
            If RemoverEspacos(CaminhoTexto & nome & "1.txt", nome) Then
                'PlaySound CaminhoSons & "converte_ok.WAV"
                PlaySound CaminhoSons & "sobre.WAV"
                'MsgBox " Arquivo convertido com sucesso "
                FrmPrincipal.Text1.Text = ""
                End If
            Else
                PlaySound CaminhoSons & "not_find.WAV"
                MsgBox "O caminho nao existe!", 48, "Tente novamente"
                FrmPrincipal.Text1.Text = ""
            End If
        End If
```

```
End If
End Sub

Private Sub Form_Load()

    Left = (Screen.Width - Width) / 2
    TxtNovoNome.Text = ""
    Label2.Visible = False
    TxtNovoNome.Visible = False

End Sub

Private Sub TxtNovoNome_KeyDown(KeyCode As Integer, Shift As Integer)
If KeyCode = 13 Then
    FileCopy FrmPrincipal.Text1.Text, Caminho & TxtNovoNome.Text & extensao
    Kill (FrmPrincipal.Text1.Text)
    If extensao = ".doc" Or extensao = ".rtf"
    Or extensao = ".xls" Or extensao = ".htm" Then
        If ConverteWordDocumento(Caminho &
            TxtNovoNome.Text & extensao, wdFormatText,
            CaminhoTexto & TxtNovoNome.Text & "1.txt") Then
            If RemoverEspacos(CaminhoTexto & TxtNovoNome.Text &
                "1.txt", TxtNovoNome.Text) Then
                PlaySound CaminhoSons & "converte_ok.WAV"
                MsgBox " Arquivo convertido com sucesso ! "
                TxtNovoNome.Text = ""
            End If
        Else
            PlaySound CaminhoSons & "not_find.WAV"
            MsgBox "O caminho não existe!", 48, "Tente novamente"
            TxtNovoNome.Text = ""
        End If
    End If
    FrmPrincipal.Text1.Text = ""
    FrmPrincipal.Text1.SetFocus
    Unload Me
End If

End Sub

Private Sub TxtNovoNome_KeyPress(KeyAscii As Integer)
Label1.Caption = KeyAscii
If X = True Then
    Text1.Text = ""

```

```

End If
If Label1.Caption = "48" Then
    letra = "0"
    PlaySound CaminhoSons & "0.WAV"
End If
If Label1.Caption = "49" Then
    letra = "1"
    PlaySound CaminhoSons & "1.WAV"
End If
If Label1.Caption = "50" Then
    letra = "2"
    PlaySound CaminhoSons & "2.WAV"
End If

```

A.3 Módulo Converte

Function ConverteWordDocumento(ByVal sArquivoNome As String, Optional ByVal wdFormato As WdSaveFormat = wdFormatText, Optional ByVal sNovoArquivo As String) As Boolean

Dim iPointer As MousePointerConstants

Dim sExtensao As String

Dim oWord As New word.Application

On Error GoTo Trataerro iPointer = Screen.MousePointer

oWord.Documents.Open sArquivoNome, False, False, False, , , , , - wdOpenFormatAuto

If Len(sNovoArquivo) = 0 Then

sNovoArquivo = sArquivoNome

If InStr(sNovoArquivo, ".") > 0 Then

sNovoArquivo = Left\$(sNovoArquivo, InStr(sNovoArquivo, ".") - 1)

sExtensao = Switch(wdFormato = wdFormatDocument, ".doc", -

wdFormato = wdFormatDOSText, ".txt", -

wdFormato = wdFormatDOSTextLineBreaks, ".txt", -

wdFormato = wdFormatEncodedText, ".txt", -

wdFormato = wdFormatHTML, ".htm", -

wdFormato = wdFormatRTF, ".rtf", -

wdFormato = wdFormatTemplate, ".doc", -

wdFormato = wdFormatText, ".dot", -

wdFormato = wdFormatTextLineBreaks, ".txt", -

```

        wdFormato = wdFormatUnicodeText, “.txt”)

        sArquivoNome = sArquivoNome & sExtensao
    End If
End If
oWord.ActiveDocument.SaveAs sNovoArquivo, wdFormato, , , False
oWord.Documents(sNovoArquivo).Close
oWord.Quit
Set oWord = Nothing

ConverteWordDocumento = True

Trataerro:
Screen.MousePointer = iPointer
End Function

```

```
Function RemoverEspaces(sNomeArquivo As String, NomeN As String)
```

```

Dim sNovoArquivo As String
Dim Dados As String * 20
Dim Dados1 As String
Dim X As Integer
Dim d1 As String
Dim d2 As String
Dim arqlivre As Integer
Dim Cont As Byte

```

```

sNovoArquivo = NomeN & “.txt”
Open sNomeArquivo For Random As #1 Len = 20
Open CaminhoTexto & sNovoArquivo For Binary As #2

```

```

While Not EOF(1)
    Get #1, , Dados
    For X = 1 To 19
        d1 = Mid(Dados, X, 1)
        d2 = Mid(Dados, X + 1, 1)
        If Asc(d1) < 32 Then d1 = Chr(32)
        If Asc(d2) < 32 Then d2 = Chr(32)
        If d1 = d2 Then

```

```
    If d1 = Chr(10) Then
        d1 = ""
    ElseIf d1 = Chr(32) Then
        d1 = ""
    ElseIf d1 = Chr(13) Then
        d1 = ""
    End If
End If
Dados1 = Dados1 & d1
Next X
d1 = Mid(Dados, 20, 1)
Dados1 = Dados1 & d1
Put #2, , Dados1
Dados1 = ""
Wend
RemoverEspacos = True
Close #1
Close #2
Kill (sNomeArquivo)
```

End Function

A.4 Módulo de declarações de variáveis (Declarar)

```
Global QuantCel As String
Global X As Boolean
Global letra As String
Global extensao As String
Global nome As String
Global q As Integer
Global p As String
Global bot_cel As String
Global anterior As String
Global Ind As Integer
Global Aux As Integer
Global Te1 As String
Global Matriz() As String
Global Linha As String
Global CaminhoSons As String
Global CaminhoTexto As String
Global Caminho As String
```



```
Public Function FileExist(FileName As String) As Boolean
```

```
    Dim TempAttr As Integer  
    On Error GoTo ErrorFileExist  
    TempAttr = GetAttr(FileName)  
    FileExist = True  
    GoTo ExitFileExist
```

```
ErrorFileExist:  
    FileExist = False  
    Resume ExitFileExist
```

```
ExitFileExist:  
    On Error GoTo ExitFileExist  
End Function
```

A.5 Módulo que executa os sons (FunçãoPlaySound)

```
Public Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" (ByVal  
    lpszSoundName As String,  
    ByVal uFlags As Long) As Long
```

```
Public Enum SndPlayFlags  
    SND_SYNC = &H0  
    SND_ASYNC = &H1  
    SND_NODEFAULT = &H2  
    SND_MEMORY = &H4  
    SND_LOOP = &H8  
    SND_NOSTOP = &H10 End Enum
```

```
Public Sub PlaySound(File As String, Optional Flags _ As SndPlayFlags)
```

```
    If File = "" Then  
        Call sndPlaySound(0&, 0)  
    Else  
        Call sndPlaySound(File, Flags)  
    End If  
End Sub
```

APÊNDICE B – Rotinas principais do Software Embarcado

Assim como no Visual Basic, muitas rotinas são repetitivas e não serão apresentadas por inteiro. O que é válido para um caractere é válido para todos os outros. Portanto, as rotinas a seguir estão incompletas.

B.1 Declarações

```
#DEFINE ICD = TRUE
INCLUDE "P16F877.INC"
ERRORLEVEL -302
ERRORLEVEL -307
_CONFIG _WDT_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF
```

```
BANK0 MACRO
    BCF STATUS,RP0
    BCF STATUS,RP1
    BCF STATUS,IRP
ENDM
```

```
BANK1 MACRO
    BSF STATUS,RP0
    BCF STATUS,RP1
    BCF STATUS,IRP
ENDM
```

```
BANK2 MACRO
    BCF STATUS,RP0
    BSF STATUS,RP1
    BSF STATUS,IRP
ENDM
```

BANK3 MACRO

BSF STATUS,RP0

BSF STATUS,RP1

BSF STATUS,IRP

ENDM

PUSH MACRO

MOVWF SALVA_W

SWAPF STATUS,W

CLRF STATUS

MOVWF SALVA_STATUS

MOVF PCLATH,W

MOVWF SALVA_PCLATH

CLRF PCLATH

MOVF FSR,W

MOVWF SALVA_FSR

ENDM

PULL MACRO

MOVF SALVA_FSR,W

MOVWF FSR

MOVF SALVA_PCLATH,W

MOVWF PCLATH

SWAPF SALVA_STATUS,W

MOVWF STATUS

SWAPF SALVA_W,F

SWAPF SALVA_W,W

ENDM

DISABLE_IRQ MACRO

STOP_INT

BCF INTCON,GIE

BTFSC INTCON,GIE

GOTO STOP_INT

ENDM

ENABLE_IRQ MACRO

BSF INTCON,GIE

BTFSS INTCON,GIE

GOTO \$-2

ENDM

FIM_TEXTO EQU B'00000001'

CLEAR EQU B'11111111'

ALFABETIZA EQU B'11111110'

TEXTO EQU B'11111101'

```
#DEFINE PONTO1 PORTC,0
#DEFINE PONTO2 PORTC,1
#DEFINE PONTO3 PORTC,2
#DEFINE PONTO4 PORTC,3
#DEFINE PONTO5 PORTD,0
#DEFINE PONTO6 PORTD,1
#DEFINE PONTO7 PORTD,2
#DEFINE LED1 PORTD,3
#DEFINE CELA1 PORTB,2
#DEFINE CELA2 PORTB,3
#DEFINE CELA3 PORTB,4
#DEFINE CELA4 PORTB,5
#DEFINE CELA5 PORTE,0
#DEFINE CELA6 PORTE,1
#DEFINE CELA7 PORTD,4
#DEFINE CELA8 PORTD,5
#DEFINE MENU PORTA,0
#DEFINE AVANCAR PORTA,1
#DEFINE RECUAR PORTA,2
#DEFINE OK PORTA,3
#DEFINE BUZZER PORTE,2
#DEFINE BIT_ALFABETIZA CONTROLE_CENTRAL,0
#DEFINE BIT_TEXTO CONTROLE_CENTRAL,1
#DEFINE BIT_TIMER1 CONTROLE_CENTRAL,2
#DEFINE BIT_SELECAO CONTROLE_CENTRAL,3
#DEFINE BIT_CELA1 AUXBIT1,0
#DEFINE BIT_CELA2 AUXBIT1,1
#DEFINE BIT_CELA3 AUXBIT1,2
#DEFINE BIT_CELA4 AUXBIT1,3
#DEFINE BIT_CELA5 AUXBIT1,4
#DEFINE BIT_CELA6 AUXBIT1,5
#DEFINE BIT_CELA7 AUXBIT1,6
#DEFINE BIT_CELA8 AUXBIT1,7
#DEFINE BIT_CELA_ATIVA1 AUXBIT2,0
#DEFINE BIT_CELA_ATIVA2 AUXBIT2,1
#DEFINE BIT_CELA_ATIVA3 AUXBIT2,2
#DEFINE BIT_CELA_ATIVA4 AUXBIT2,3
#DEFINE BIT_CELA_ATIVA5 AUXBIT2,4
#DEFINE BIT_CELA_ATIVA6 AUXBIT2,5
#DEFINE BIT_CELA_ATIVA7 AUXBIT2,6
#DEFINE BIT_CELA_ATIVA8 AUXBIT2,7

CBLOCK 0X20
    SALVA_W
```

```
SALVA_PCLATH
SALVA_FSR
SALVA_STATUS
AUX1
AUX2
AUX3
DADO
COUNT_BIT
UARTOCK
CONT_VAR
CONTROLE_CENTRAL
AUXBIT1
AUXBIT2
AUXBIT3
AUXCEL1
AUXCEL2
AUXCEL3
AUXCEL4
AUXCEL5
AUXCEL6
AUXCEL7
AUXCEL8
AUXEXIBIR1
AUXEXIBIR2
AUXEXIBIR3
AUXEXIBIR4
AUXEXIBIR5
AUXEXIBIR6
AUXEXIBIR7
AUXEXIBIR8
TIMER1
AUX_PORTAS
ENDC
```

B.2 Rotina Principal

```
ENABLE_IRQ
ROTINA_PRINCIPAL
    BSF LED1
    CALL DELAY_350_MILI
    BCF LED1
    CALL DELAY_350_MILI
```

```
BTFSS BIT_TEXTO
GOTO CONTINUA_ROTINA_PRINCIPAL
BTFSC BIT_SELECAO
GOTO EXIBE_PROXIMOS_DADOS
MOVF AUXEXIBIR1,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA1
CALL uDELAY_LOCAL
BCF CELA1
MOVF AUXEXIBIR2,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA2
CALL uDELAY_LOCAL
BCF CELA2
MOVF AUXEXIBIR3,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA3
CALL uDELAY_LOCAL
BCF CELA3
MOVF AUXEXIBIR4,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA4
CALL uDELAY_LOCAL
BCF CELA4
MOVF AUXEXIBIR5,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA5
CALL uDELAY_LOCAL
BCF CELA5
MOVF AUXEXIBIR6,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA6
CALL uDELAY_LOCAL
BCF CELA6
MOVF AUXEXIBIR7,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
```

```
BSF CELA7
CALL uDELAY_LOCAL
BCF CELA7
MOVF AUXEXIBIR8,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA8
CALL uDELAY_LOCAL
BCF CELA8
GOTO CONTINUA_ROTINA_PRINCIPAL
```

```
EXIBE_PROXIMOS_DADOS
MOVF AUXEXIBIR9,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA1
CALL uDELAY_LOCAL
BCF CELA1
MOVF AUXEXIBIR10,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA2
CALL uDELAY_LOCAL
BCF CELA2
MOVF AUXEXIBIR11,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA3
CALL uDELAY_LOCAL
BCF CELA3
MOVF AUXEXIBIR12,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA4
CALL uDELAY_LOCAL
BCF CELA4
MOVF AUXEXIBIR13,W
MOVWF AUX_PORTAS
CALL ASCII_CELA_LOCAL
BSF CELA5
CALL uDELAY_LOCAL
BCF CELA5
MOVF AUXEXIBIR14,W
MOVWF AUX_PORTAS
```

```
CALL ASCII.CELA.LOCAL
BSF CELA6
CALL uDELAY.LOCAL
BCF CELA6
MOVF AUXEXIBIR15,W
MOVWF AUX_PORTAS
CALL ASCII.CELA.LOCAL
BSF CELA7
CALL uDELAY.LOCAL
BCF CELA7
MOVF AUXEXIBIR16,W
MOVWF AUX_PORTAS
CALL ASCII.CELA.LOCAL
BSF CELA8
CALL uDELAY.LOCAL
BCF CELA8
```

CONTINUA_ROTINA_PRINCIPAL

```
BTFSS BIT_TIMER1
GOTO ROTINA_PRINCIPAL
DECFSZ TIMER1,F
GOTO ROTINA_PRINCIPAL
BCF BIT_TIMER1
MOVLW 30
MOVWF TIMER1
CALL LIMPA_CELA_FULL.LOCAL
GOTO ROTINA_PRINCIPAL
```

B.3 Conversão Braille

COMPARA_ALFABETIZA

```
MOVF UARTOCK,W
XORLW 'a'
BTFSS STATUS,Z
GOTO $+2
GOTO A_MINUSCULO
MOVF UARTOCK,W
XORLW 'b'
BTFSS STATUS,Z
GOTO $+2
GOTO B_MINUSCULO
MOVF UARTOCK,W
```



```
XORLW 'c'  
BTFSS STATUS,Z  
GOTO $+2  
GOTO C_MINUSCULO  
MOVF UARTOCK,W  
XORLW 'd'  
BTFSS STATUS,Z  
GOTO $+2  
GOTO D_MINUSCULO  
MOVF UARTOCK,W  
XORLW 'e'  
BTFSS STATUS,Z  
GOTO $+2  
GOTO E_MINUSCULO
```

MAIUSCULO

```
BTFSB BIT_TEXTO  
GOTO MAIUSCULO_TEXTO  
MOVLW B'00101000'  
MOVWF AUX_PORTAS  
CALL ASCII_CELA  
BSF CELA1  
CALL uDELAY  
BCF CELA1  
CALL uDELAY  
RETURN
```

MAIUSCULO_TEXTO

```
MOVLW B'00000001'  
MOVWF AUX_PORTAS  
CALL ACERTA_POSICAO  
RETURN
```

NUMERO

```
BTFSB BIT_TEXTO  
GOTO NUMERO_TEXTO  
MOVLW B'00111100'  
MOVWF AUX_PORTAS  
CALL ASCII_CELA  
BSF CELA1  
CALL uDELAY  
BCF CELA1  
CALL uDELAY  
RETURN
```

NUMERO_TEXTO

```
MOVLW B'00111100'  
MOVWF AUX_PORTAS  
CALL ACERTA_POSICAO  
RETURN
```

SAIDA_ACERTA_POSICAO

```
MOVLW 30  
MOVWF TIMER1  
BSF BIT_TIMER1  
MOVF UARTOCK,W  
XORLW 'a'  
BTSS STATUS,Z  
GOTO $+2  
GOTO A_MINUSCULO  
MOVF UARTOCK,W  
XORLW 'b'  
BTSS STATUS,Z  
GOTO $+2  
GOTO B_MINUSCULO  
MOVF UARTOCK,W  
XORLW 'c'  
BTSS STATUS,Z  
GOTO $+2  
GOTO C_MINUSCULO  
MOVF UARTOCK,W  
XORLW 'd'  
BTSS STATUS,Z  
GOTO $+2  
GOTO D_MINUSCULO  
MOVF UARTOCK,W  
XORLW 'e'  
BTSS STATUS,Z  
GOTO $+2  
GOTO E_MINUSCULO
```

LIMPA_CELA

```
MOVLW B'00000000'  
MOVWF AUX_PORTAS  
CALL ASCII_CELA  
BSF CELA2  
CALL uDELAY  
BCF CELA2
```

```
CALL uDELAY
RETURN
```

LIMPA_CELA_FULL

```
MOVLW B'00000000'
MOVWF AUX_PORTAS
CALL ASCII_CELA
BSF CELA1
BSF CELA2
BSF CELA3
BSF CELA4
BSF CELA5
BSF CELA6
BSF CELA7
BSF CELA8
CALL uDELAY
BCF CELA1
BCF CELA2
BCF CELA3
BCF CELA4
BCF CELA5
BCF CELA6
BCF CELA7
BCF CELA8
RETURN
```

ASCII_CELA

```
MOVLW B'11110000'
ANDWF PORTC,F
MOVLW B'11111100'
ANDWF PORTD,F
MOVF AUX_PORTAS,W
ANDLW B'00001111'
IORWF PORTC,F
SWAPF AUX_PORTAS,W
ANDLW B'00000011'
IORWF PORTD,F
RETURN
```

ACERTA_POSICAO

```
BTFSC BIT_CELA_ATIVA1
GOTO $+5
BSF BIT_CELA_ATIVA1
MOVF AUX_PORTAS,W
```

```
MOVWF AUXEXIBIR1
GOTO SAIDA_ACERTA_POSICAO
BTFSC BIT_CELA_ATIVA2
GOTO $+5
BSF BIT_CELA_ATIVA2
MOVF AUX_PORTAS,W
MOVWF AUXEXIBIR2
GOTO SAIDA_ACERTA_POSICAO
BTFSC BIT_CELA_ATIVA3
GOTO $+5
BSF BIT_CELA_ATIVA3
MOVF AUX_PORTAS,W
MOVWF AUXEXIBIR3
GOTO SAIDA_ACERTA_POSICAO
RETURN
```

A.MINUSCULO

```
BTFSC BIT_TEXTO
GOTO A_MINUSCULO_TEXTO
CALL LIMPA_CELA
MOVLW B'00000001'
MOVWF AUX_PORTAS
CALL ASCII_CELA
BSF CELA1
CALL uDELAY
BCF CELA1
GOTO RETORNO
```

A.MINUSCULO_TEXTO

```
MOVLW B'00000001'
MOVWF AUX_PORTAS
CALL ACERTA_POSICAO
GOTO RETORNO
```

B.MINUSCULO

```
BTFSC BIT_TEXTO
GOTO B_MINUSCULO_TEXTO
CALL LIMPA_CELA
MOVLW B'00000011'
MOVWF AUX_PORTAS
CALL ASCII_CELA
BSF CELA1
CALL uDELAY
BCF CELA1
```

GOTO RETORNO

B.MINUSCULO_TEXTO

```
MOVLW B'00000011'  
MOVWF AUX_PORTAS  
CALL ACERTA_POSICAO  
GOTO RETORNO
```

C.MINUSCULO

```
BTFSC BIT_TEXTO  
GOTO B_MINUSCULO_TEXTO  
CALL LIMPA_CELA  
MOVLW B'00001001'  
MOVWF AUX_PORTAS  
CALL ASCII_CELA  
BSF CELA1  
CALL uDELAY  
BCF CELA1  
GOTO RETORNO
```

C.MINUSCULO_TEXTO

```
MOVLW B'00001001'  
MOVWF AUX_PORTAS  
CALL ACERTA_POSICAO  
GOTO RETORNO
```

D.MINUSCULO

```
BTFSC BIT_TEXTO  
GOTO D_MINUSCULO_TEXTO  
CALL LIMPA_CELA  
MOVLW B'00011001'  
MOVWF AUX_PORTAS  
CALL ASCII_CELA  
BSF CELA1  
CALL uDELAY  
BCF CELA1  
GOTO RETORNO
```

D.MINUSCULO_TEXTO

```
MOVLW B'00011001'  
MOVWF AUX_PORTAS  
CALL ACERTA_POSICAO  
GOTO RETORNO
```

E.MINUSCULO

```
BTFSC BIT_TEXTO
GOTO E_MINUSCULO_TEXTO
CALL LIMPA_CELA
MOVLW B'00010001'
MOVWF AUX_PORTAS
CALL ASCII_CELA
BSF CELA1
CALL uDELAY
BCF CELA1
GOTO RETORNO
```

E_MINUSCULO_TEXTO

```
MOVLW B'00010001'
MOVWF AUX_PORTAS
CALL ACERTA_POSICAO
GOTO RETORNO
```

B.4 Outras Rotinas

LETRA_ALFABETIZA

```
LCALL COMPARA_ALFABETIZA
GOTO SAIDA_RAPIDA
```

CARACTERES_TEXTO

```
MOVF UARTOCK,W
XORLW FIM_TEXTO
BTFSS STATUS,Z
GOTO CARACTERES_TEXTO_CONTINUE
BTFSS BIT_CELA1
GOTO $+6
MOVF AUXCEL1,W
MOVWF UARTOCK
LCALL COMPARA_ALFABETIZA
BCF BIT_CELA1
BTFSS BIT_CELA2
GOTO $+6
MOVF AUXCEL2,W
MOVWF UARTOCK
LCALL COMPARA_ALFABETIZA
BCF BIT_CELA2
BTFSS BIT_CELA3
```

```
GOTO $+6
MOVF AUXCEL3,W
MOVWF UARTOCK
LCALL COMPARA_ALFABETIZA
BCF BIT_CELA3
BTFSS BIT_CELA4
GOTO $+6
MOVF AUXCEL4,W
MOVWF UARTOCK
LCALL COMPARA_ALFABETIZA
BCF BIT_CELA4
BTFSS BIT_CELA5
GOTO $+6
MOVF AUXCEL5,W
MOVWF UARTOCK
LCALL COMPARA_ALFABETIZA
BCF BIT_CELA5
BTFSS BIT_CELA6
GOTO $+6
MOVF AUXCEL6,W
MOVWF UARTOCK
LCALL COMPARA_ALFABETIZA
BCF BIT_CELA6
BTFSS BIT_CELA7
GOTO $+6
MOVF AUXCEL7,W
MOVWF UARTOCK
LCALL COMPARA_ALFABETIZA
BCF BIT_CELA7
BTFSS BIT_CELA8
GOTO $+6
MOVF AUXCEL8,W
MOVWF UARTOCK
LCALL COMPARA_ALFABETIZA
BCF BIT_CELA8
MOVLW B'00000000'
BTFSS BIT_CELA_ATIVA1
MOVWF AUXEXIBIR1
BTFSS BIT_CELA_ATIVA2
MOVWF AUXEXIBIR2
BTFSS BIT_CELA_ATIVA3
MOVWF AUXEXIBIR3
BTFSS BIT_CELA_ATIVA4
```

```
MOVWF AUXEXIBIR4
BTFSS BIT_CELA_ATIVA5
MOVWF AUXEXIBIR5
BTFSS BIT_CELA_ATIVA6
MOVWF AUXEXIBIR6
BTFSS BIT_CELA_ATIVA7
MOVWF AUXEXIBIR7
BTFSS BIT_CELA_ATIVA8
MOVWF AUXEXIBIR8
GOTO SAIDA_RAPIDA
```

CARACTERES_TEXTO_CONTINUE

```
MOVF UARTOCK,W
BTFSC BIT_CELA1
GOTO $+6
MOVWF AUXCEL1
BSF BIT_CELA1
CLRF AUXBIT2
CLRF AUXBIT3
GOTO SAIDA_RAPIDA
MOVF UARTOCK,W
BTFSC BIT_CELA2
GOTO $+4
MOVWF AUXCEL2
BSF BIT_CELA2
GOTO SAIDA_RAPIDA
MOVF UARTOCK,W
BTFSC BIT_CELA3
GOTO $+4
MOVWF AUXCEL3
BSF BIT_CELA3
GOTO SAIDA_RAPIDA
MOVF UARTOCK,W
BTFSC BIT_CELA4
GOTO $+4
MOVWF AUXCEL4
BSF BIT_CELA4
GOTO SAIDA_RAPIDA
MOVF UARTOCK,W
BTFSC BIT_CELA5
GOTO $+4
MOVWF AUXCEL5
BSF BIT_CELA5
GOTO SAIDA_RAPIDA
```



```
MOVWF UARTOCK,W
BTFSC BIT_CELA6
GOTO $+4
MOVWF AUXCEL6
BSF BIT_CELA6
GOTO SAIDA_RAPIDA
MOVWF UARTOCK,W
BTFSC BIT_CELA7
GOTO $+4
MOVWF AUXCEL7
BSF BIT_CELA7
GOTO SAIDA_RAPIDA
MOVWF UARTOCK,W
MOVWF AUXCEL8
GOTO SAIDA_RAPIDA
```

SAIDA_RAPIDA

```
BCF PIR1,RCIF
NOP
BCF PIR1,RCIF
BSF PIE1,RCIE
NOP
GOTO INT_SAIDA
```

LIMPA_CELA_FULL_LOCAL

```
BCF PONTO1
BCF PONTO2
BCF PONTO3
BCF PONTO4
BCF PONTO5
BCF PONTO6
BSF CELA1
BSF CELA2
BSF CELA3
BSF CELA4
BSF CELA5
BSF CELA6
BSF CELA7
BSF CELA8
CALL DELAY_100_MILI
BCF CELA1
BCF CELA2
BCF CELA3
BCF CELA4
```

```
BCF CELA5
```

```
BCF CELA6
```

```
BCF CELA7
```

```
BCF CELA8
```

```
RETURN
```

```
ASCII_CELA_LOCAL
```

```
MOVLW B'11110000'
```

```
ANDWF PORTC,F
```

```
MOVLW B'11111100'
```

```
ANDWF PORTD,F
```

```
MOVF AUX_PORTAS,W
```

```
ANDLW B'00001111'
```

```
BCF CELA1
```

```
BCF CELA2
```

```
BCF CELA3
```

```
BCF CELA4
```

```
BCF CELA5
```

```
BCF CELA6
```

```
BCF CELA7
```

```
BCF CELA8
```

```
IORWF PORTC,F
```

```
SWAPF AUX_PORTAS,W
```

```
ANDLW B'00000011'
```

```
IORWF PORTD,F
```

```
RETURN
```

```
uDELAY_LOCAL
```

```
MOVLW D'100'
```

```
MOVWF AUX1
```

```
uDELAY_LOCAL_SX
```

```
DECFSZ AUX1,f
```

```
GOTO uDELAY_LOCAL_SX
```

```
RETURN
```

```
DELAY_100_MILI
```

```
MOVLW D'250'
```

```
MOVWF AUX1
```

```
MOVLW D'10'
```

```
MOVWF AUX2
```

```
MD
```

```
GOTO $+1
```

```
NOP
```

```
GOTO $+1
DECFSZ AUX1,F
GOTO MD
NOP
GOTO $+1
MOVLW D'250'
MOVWF AUX1
DECFSZ AUX2,F
GOTO MD
RETURN
```

DELAY_350_MILI

```
CLRF DADO
```

DELAY1

```
INCF DADO,F
BTFSC DADO,3
RETURN
MOVLW 0X0F
MOVWF AUX1
```

REPETE1

```
MOVLW 0XF9
MOVWF AUX2
```

REPETE2

```
NOP
DECFSZ AUX2,F
GOTO REPETE2
DECFSZ AUX1,F
GOTO REPETE1
GOTO DELAY1
```

INI_VARIAVEIS

```
CLRF COUNT_BIT
CLRF CONTROLE_CENTRAL
CLRF AUXBIT1
CLRF AUXBIT2
CLRF AUXBIT3
CLRF AUXCEL1
CLRF AUXCEL2
CLRF AUXCEL3
CLRF AUXCEL4
CLRF AUXCEL5
CLRF AUXCEL6
CLRF AUXCEL7
CLRF AUXCEL8
```

```
CLRF AUXEXIBIR1
CLRF AUXEXIBIR2
CLRF AUXEXIBIR3
CLRF AUXEXIBIR4
CLRF AUXEXIBIR5
CLRF AUXEXIBIR6
CLRF AUXEXIBIR7
CLRF AUXEXIBIR8
RETURN
```

B.5 Comunicação Serial

UARTINI

```
BANK1
BSF TRISB,1
BSF TXSTA,CSRC
BCF TXSTA,TX9
BSF TXSTA,TXEN
BCF TXSTA,SYNC
BCF TXSTA,BRGH
MOVLW D'25'
MOVWF SPBRG
BSF PIE1,RCIE
BANK0
BSF RCSTA,SPEN
BCF RCSTA,RX9
BSF RCSTA,SREN
BSF RCSTA,CREN
BCF RCSTA,ADDEN
MOVF RCREG,w
MOVF RCREG,w
MOVF RCREG,w
BSF INTCON,PEIE
RETURN
```

SERIAL.INTERUPT

```
MOVF RCREG,W
MOVWF UARTOCK
MOVF UARTOCK,W
XORLW CLEAR
BTFS STATUS,Z
GOTO $+6
```

```
CLRF CONTROLE_CENTRAL
CLRF AUXBIT1
CLRF AUXBIT2
CLRF AUXBIT2
GOTO SAIDA_RAPIDA
BTFSC BIT_ALFABETIZA
GOTO LETRA_ALFABETIZA
BTFSC BIT_TEXTO
GOTO CARACTERES_TEXTO
MOVF UARTOCK,W
XORLW ALFABETIZA
BTFSS STATUS,Z
GOTO $+3
BSF BIT_ALFABETIZA
GOTO SAIDA_RAPIDA
MOVF UARTOCK,W
XORLW TEXTO
BTFSS STATUS,Z
GOTO SAIDA_RAPIDA
BSF BIT_TEXTO
GOTO SAIDA_RAPIDA
```

APÊNDICE C – Manual do SoftBraille

C.1 Área visual do programa

Uma vez que o programa, aqui desenvolvido, seja executado no microcomputador, a tela da Figura ?? estará presente no monitor. Esta tela é composta de três caixas de texto e sete botões. Uma vez que esta imagem não pode ser vista pelo DV, cada botão tem um atalho no teclado. Os dizeres: caixa de texto 1, caixa de texto 2 e caixa de texto 3 não fazem parte da tela, estando ali presentes para facilitar a compreensão. O programa também não conta com várias telas e menus, pois isto dificultaria a operação do *software* por parte do DV. As caixas de texto, são utilizadas pelos videntes¹ que possam estar utilizando o programa, ou auxiliando o DV na utilização.

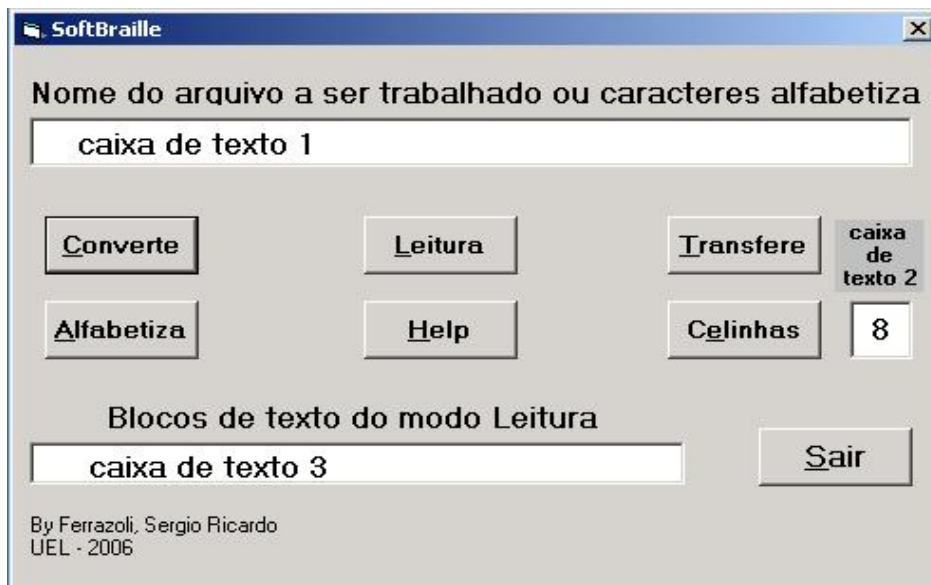


Figura C.1: Tela principal do programa SoftBraille.

C.1.1 Os botões, sua teclas de atalho e suas funções detalhadas

- **Converte:** este botão tem a função de converter um arquivo que possua formatação, em um simples arquivo texto. Este arquivo será salvo automaticamente, após convertido em uma pasta do programa específica para guardar os arquivos convertidos,

¹Pessoas que possuem visão normal.



Figura C.2: Tela auxiliar do botão converte.



Figura C.3: Complemento da tela auxiliar do botão converte na opção renomear.

a fim de facilitar a sua localização pelo DV. O atalho no teclado para este botão é a tecla Alt juntamente com a tecla C. No caso de o nome do arquivo a ser convertido já existir na pasta do sistema, uma nova tela surgirá como nas Figuras ?? e ??.

- **Leitura:** este botão tem a função de abrir um arquivo já convertido e enviá-lo para a Interface Braille Automatizada em blocos de caracteres de tamanho igual ao número de celinhas que se tenha na interface. O atalho no teclado para este botão é a tecla Alt juntamente com a tecla L. Assim, depois de selecionada esta opção, o programa SoftBraille começa a percorrer o texto em blocos de caracteres igual ao de celinhas e envia estes blocos para a interface. Para adiantar um bloco de caracteres ou retroceder, o DV se utiliza das setas para baixo e para cima, do teclado do microcomputador, respectivamente.
- **Transfere:** este botão tem a função de abrir um arquivo já convertido e enviá-lo totalmente, e de uma vez, para a Interface Braille Automatizada. Desta forma, a interface pode ser desligada do Microcomputador e repetirá sempre o mesmo texto, quando for solicitado pelo acionamento de seus botões de controle. O atalho no teclado para este botão é a tecla Alt juntamente com a tecla T.
- **Sair:** este botão tem a função de liberar a porta serial e fechar o programa. O atalho no teclado para este botão é a tecla Alt juntamente com a tecla S.
- **Help:** este é o botão de ajuda que informa resumidamente ao usuário do *software* as funções básicas dos botões do programa. Esta informação se dá por meio de um arquivo de áudio gravado no microcomputador. O atalho no teclado para este botão é a tecla Alt juntamente com a tecla H.

- **Celinhas:** este é o botão que define a quantidade de celinhas que estão à disposição para uso na Interface Braille Automatizada. Dessa forma, é possível dimensionar o tamanho da interface quanto à quantidade de caracteres que serão mostrados simultaneamente e portanto a quantidade de caracteres que o SoftBraille deverá enviar à interface simultaneamente no modo de Leitura. O atalho no teclado para este botão é a tecla Alt juntamente com a tecla E.
- **Alfabetiza:** este botão tem a função de enviar à interface o caractere digitado no teclado. Neste módulo do sistema, serão utilizadas apenas uma ou duas celinhas dependendo do caso. Isto se deve ao fato de que um caractere pode ocupar até duas celinhas dependendo do caso, como visto na seção 2.2.1, e na próxima tecla pressionada ele substitui nas mesmas celinhas o novo caractere, utilizando assim sempre a primeira e segunda celinha quando necessário.

Podemos acrescentar um novo botão, Status, que ao serem acionadas juntamente as teclas ALT e U, relata em audio o histórico dos últimos comandos dados pelo DV.

É importante citar que o termo Alfabetização não significa o ato de ensinar um DV a ler e escrever, mas sim de fazer que o mesmo se familiarize com o alfabeto, neste caso representado em Braille.

C.1.2 As caixas de texto e suas funções

- A caixa de texto 1, como mostrada na Figura ??, é onde será digitado o nome do arquivo a ser trabalhado nos botões Converte, Leitura e Transmite ou onde serão digitados os caracteres no caso do modo Alfabetiza. Quando o DV vai digitar o arquivo para utilizar a opção converte, ele necessita colocar o local onde o arquivo está, o nome do arquivo e a sua extensão. Exemplo: (A:/trabalho.doc) isto significa que o arquivo está no drive A, e tem o nome de trabalho e sua extensão é doc. Assim, é necessário que o DV conheça estes dados antecipadamente.
- A caixa de texto 2, como mostrada na Figura ??, é onde será digitado a quantidade de celinhas disponíveis no *hardware*. Isto se deve ao fato de o sistema poder contar com números diferentes de celinhas, de acordo com a quantidade que o usuário desejar adquirir. Assim, o DV seleciona a quantidade de celinhas necessárias e o programa se reconfigura. Não é necessário que o DV selecione isto todas as vezes que for utilizar o programa, pois ele grava esta quantidade e toda vez que for executado, estará com a ultima quantidade escolhida. Esta opção é interessante pois como há a possibilidade de se colocar mais ou menos celinhas, numa possível ampliação, o próprio DV pode selecionar a nova quantidade necessária.
- A caixa de texto 3, como mostrada na Figura ??, é o local que conterà o bloco de caracteres da opção de leitura. Claro que o DV não se utiliza desta caixa, porém é interessante para um vidente acompanhar o que está sendo emitido para a interface, assim, os caracteres que estiverem nesta caixa são exatamente os caracteres representados na interface em Braille.

C.2 Funcionamento do software

O *software*, aqui desenvolvido chamado SoftBraille, é destinado para ser utilizado por um DV em ambiente Windows, da Microsoft. Como dito anteriormente, foi desenvolvido em apenas uma janela contendo botões que disparam os eventos e, como o DV não pode ver os botões, contará com teclas de atalho para facilitar sua utilização. Assim, quando o *software* for executado pelo DV ele poderá ser totalmente utilizado sem a necessidade do mouse. Durante a instalação do *software* no microcomputador do DV, será criada uma pasta que conterà todos os arquivos necessários para sua execução. Quando o DV digitar os caracteres no teclado, o microcomputador que possuir uma placa de som e auto-falantes, executará um arquivo que conterà o som relativo àquele caractere e assim, o microcomputador irá falar a tecla digitada para o usuário.



Figura C.4: Exemplo de um arquivo original com formatações e figura.

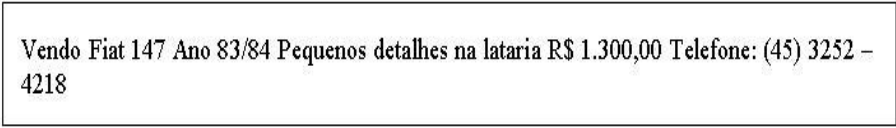
O seu funcionamento pode ser resumido da seguinte forma:

- Quando o usuário executar o programa, uma gravação executa a seguinte mensagem: Bem-vindo ao SoftBraille. Digite a opção ou tecla ALT H para ajuda. Neste momento o cursor estará na caixa de texto 1, Figura ??, e o programa aguarda algo ser digitado ou alguma tecla ser selecionada.

- Uma vez que o nome de um arquivo é digitado na caixa de texto 1, após a execução do programa, pode-se optar por “Converte” o arquivo, “Leitura” do arquivo ou “Transfere” o arquivo. Cada um deles é explicado a seguir:
 - Converte: nesta função, deve-se digitar o arquivo completo, incluindo a sua extensão. Por exemplo: A:/vendo.doc, como mostrado na Figura ??, e a seguir pressionar juntamente as teclas ALT e C. Neste momento o programa irá retirar a formatação do arquivo e salvar o documento “vendo” dentro de uma pasta do sistema já como arquivo texto simples sem formatação alguma, como mostrado na Figura ?. Notar que nesta opção, é necessário saber o diretório raiz do arquivo (A:/), o nome do arquivo (vendo) e a extensão do arquivo (.doc).
 - Leitura: para o usuário efetuar uma leitura em Braille, digita-se, na caixa de texto 1, o nome do arquivo e pressionam-se juntamente as teclas ALT e L. Neste momento o *software* verifica a quantidade de celinhas selecionadas e divide o arquivo texto em pequenos vetores de caracteres igual ao de celinhas existentes. Assim, envia para a interface pacotes de caracteres. Estes caracteres são mostrados também na caixa de texto 3, Figura ?. A interface por sua vez converte para o respectivo código Braille e representa estes códigos na Interface Braille Automatizada. Com as setas no teclado do computador para a esquerda e para a direita pode-se enviar os blocos seguintes e anteriores, proporcionando assim a leitura do documento. Nesta opção, não é necessário digitar o caminho e a extensão do arquivo, pois este já foi convertido e está salvo numa pasta do sistema do programa SoftBraille denominada textos.
 - Transfere: A interface possui um modo de operação independente do computador. Para isso é necessário que se envie um arquivo texto para ser gravado na memória da interface. Isto pode ser feito digitando-se o nome do arquivo e pressionando-se simultaneamente as teclas ALT e T. Assim, este arquivo fica gravado na interface e o cabo serial pode ser desligado da mesma. Com isso o texto gravado pode ser lido pressionando-se os botões de controle da interface que serão melhor detalhados no *Hardware*. Nesta opção, não é necessário se digitar o caminho e a extensão do arquivo, pois este já foi convertido e está salvo numa pasta do sistema do programa SoftBraille denominada textos.
- Caso o usuário opte por ajuda, ele pressiona juntamente as teclas ALT e H, atalho para o botão Help. O programa executará a seguinte mensagem: Para o módulo Alfabetiza, tecla ALT A; para converter um documento digite o caminho completo do arquivo, incluindo a sua extensão e tecla ALT C; para ler um documento digite o nome do arquivo e tecla ALT L; para enviar um arquivo completo para a interface digite o nome do arquivo e tecla ALT T; para selecionar a quantidade de celinhas tecla ALT E, digite a quantidade de celinhas desejada e a seguir pressione enter; para sair tecla ALT S; para repetir a ajuda tecla ALT H.
- O programa, na sua instalação, é predefinido com a quantidade de oito celinhas. Porém, esta quantidade pode ser alterada de acordo com a necessidade individual

de cada usuário. Caso o usuário necessite fazer esta alteração, ele pressiona simultaneamente as teclas ALT e E e digita a nova quantidade de celinhas. Após a quantidade digitada, e pressionada a tecla enter, o programa volta para o estado inicial aguardando que se digite algo na caixa de texto 1, Figura ??, ou que se selecione algum botão, quer seja pelo mouse, quer pelas teclas de atalho.

- Para sair do programa, pressionam-se simultaneamente as teclas ALT e S. Com isso o *software* toca o som de uma campainha “din don” para informar ao usuário DV que o programa foi encerrado.
- O módulo alfabetiza, quando selecionado pressionando-se simultaneamente as teclas ALT e A, enviará caractere por caractere à interface. Assim, se for pressionado a tecla A, este caracter será mostrado na caixa de texto 1, e será enviado à interface que mostrará na Interface Braille Automatizada este mesmo caractere em Braille. Se em seguida for pressionada a tecla B, imediatamente apaga-se o que havia na caixa de texto e na interface e os seus conteúdos são substituídos pela nova tecla. Isto se repete até que se deseje terminar a execução. Então é necessário pressionar simultaneamente as teclas ALT e S para sair do programa.



Vendo Fiat 147 Ano 83/84 Pequenos detalhes na lataria R\$ 1.300,00 Telefone: (45) 3252 - 4218

Figura C.5: Exemplo de arquivo convertido em formato texto pronto a ser enviado para a interface.

C.2.1 Opção Converte

Foi explorada um pouco mais a opção Converte devido ao fato de esta opção possuir algumas peculiaridades. No momento da conversão, o SoftBraille irá verificar se o nome do arquivo já existe na pasta do sistema, e se existir abrirá a tela da Figura ?. Com isso o usuário irá decidir se irá renomear o arquivo, sobrescrever o arquivo ou cancelar a operação.

No caso de o usuário escolher a opção de renomear, o SoftBraille irá alterar a tela, como na Figura ?, e uma caixa de texto estará aguardando o novo nome e após teclar Enter no teclado, irá renomear o arquivo de origem, e posteriormente de modo automático irá converter o arquivo normalmente como já mencionado.

Caso o usuário deseje sobrescrever o arquivo, o SoftBraille irá gravar o novo arquivo em cima do arquivo anterior, apagando assim o mesmo e permanecendo o novo.

Na opção de cancelar, o programa volta para a tela inicial, como quando fora iniciado.

*ANEXO A – Poster apresentado na
Semana de Ciência e
Tecnologia da UEL*

Página a ser substituída pelo anexo.

*ANEXO B – Declaração de participação
no I Seminário de Inserção
Social - Educação e Trabalho
das Pessoas com Deficiência
Visual*

Página a ser substituída pelo anexo.

ANEXO C - Cópia do Depósito de Patente

Página a ser substituída pelo anexo.

Página a ser substituída pelo anexo.

Página a ser substituída pelo anexo.

*ANEXO D - Perguntas feitas a usuários
Deficientes Visuais e
Videntes durante o I
Seminário de Inserção Social
- Educação e Trabalho das
Pessoas com Deficiência
Visual.*

Página a ser substituída pelo anexo.

Página a ser substituída pelo anexo.

Página a ser substituída pelo anexo.

ANEXO E - Resumo impresso em Braille

Página a ser substituída pelo anexo.