

## TELA INTERATIVA PARA PROJETOR DE IMAGENS

EDUARDO BORGES DA COSTA



#### TELA INTERATIVA PARA PROJETOR DE IMAGENS

#### EDUARDO BORGES DA COSTA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito de Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Analisador (1): Alexandre Charles Cassiano

ASSIS 2009



#### EDUARDO BORGES DA COSTA

#### TELA INTERATIVA PARA PROJETOR DE IMAGENS

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Área de Concentração: Informática

Assis 2009

Dedico este trabalho a minha família, em especial, a minha companheira Vanderleia e meu filho Eric, que foram meu braço forte para que este trabalho fosse realizado.

#### **AGRADECIMENTOS**

Á Deus, pois sem sua vontade eu nunca estaria aqui, e graças a ele conquistamos mais esta batalha;

Á Professora Marisa Atsuko Nitto, pela orientação e dedicação que teve comigo não só neste trabalho final, mas durante todo o período acadêmico, sua ajuda será lembrada por toda a minha vida;

Á Fundação Educacional do Município de Assis, por ter me possibilitado a realização deste trabalho;

Á minha família, por estar ao meu lado todo este tempo, em especial ao meu pai Nivaldo, pelo apoio e incentivo, além das oportunas sugestões nas questões de confecção da estrutura física do projeto. Ao meu querido filho Eric, que me inspirava com sua imensa energia de criança;

Á Vanderleia, minha grande companheira, por me incentivar e dar forças para que eu conseguisse me manter na linha certa para que este trabalho fosse concluído;

Á todos os amigos, que de uma maneira ou outra estiveram presentes nos momentos mais difíceis.

**RESUMO** 

As telas touch screen tem aparecido com mais freqüência no

mercado nestes últimos anos, porém esta é constituída de um

sensor e um monitor, não sendo popularmente conhecida a venda

apenas do sensor para aplicações domésticas ou para pequenas

empresas.

Este trabalho consiste em desenvolver um sensor touch

screen que servirá para aplicações em projetores de imagens

ligados a um computador. A estrutura física deste sensor será

no formato de uma moldura, onde o projetor fará a projeção da

imagem dentro da área interna da moldura.

Será empregada uma técnica inédita na construção desta

tela touch screen, aplicando lasers na captura dos toques na

tela.

Palavras-chaves: touch screen, sensor ótico, laser.

**ABSTRACT** 

The touch screen has appeared more frequently in the

market in recent years, however is consists of a sensor and a

monitor, is not commonly known to sell just the sensor for

domestic applications or for small businesses.

This work is to develop a sensor touch screen that served

for applications in image projectors connected to a computer.

The physical structure of this sensor will be in the form of a

frame, where the projector will make a projection of the image

within the inner area of the frame. Be employed a

technique in the construction of the touch screen, using

lasers to capture the touch screen.

Keywords: touch screen, optical sensors, laser.

# LISTA DE ILUSTRAÇÕES

| Figura  | 1.  | Mouse   | 5  |
|---------|-----|---|----|
| Figura  | 2.  | Teclado                                       | 5  |
| Figura  | 3.  | Alguns softwares de reconhecimento de voz     | 6  |
| Figura  | 4.  | Conectores USB                                | 8  |
| Figura  | 5.  | Diagrama de blocos                            | 13 |
| Figura  | 6.  | Fluxograma de gravação de dados na memória de |    |
| program | na  |   | 16 |
| Figura  | 7.  | Tela do compilador PCWH                       | 19 |
| Figura  | 8.  | Criando um arquivo fonte                      | 19 |
| Figura  | 9.  | Salvando o arquivo                            | 20 |
| Figura  | 10. | Código de exemplo                             | 20 |
| Figura  | 11. | Compilando um arquivo                         | 21 |
| Figura  | 12. | Relatório de compilação                       | 21 |
| Figura  | 13. | Resistores fixos comuns                       | 23 |
| Figura  | 14. | Capacitores: símbolos e aspectos              | 24 |
| Figura  | 15. | Transistores bipolares                        | 25 |
| Figura  | 16. | Aspectos dos transistores                     | 26 |
| Figura  | 17. | Transistor montado em radiador de calor       | 26 |
| Figura  | 18. | Diodos  | 27 |
| Figura  | 19. | Diodo Laser                                   | 27 |
| Figura  | 20. | Receptor infravermelho                        | 28 |
| Figura  | 21. | Circuito integrado                            | 29 |
| Figura  | 22. | Ícone do compilador PCWH                      | 30 |
| Figura  | 23. | IC-Prog                                       | 30 |

| Figura | 24. | Interface de programação PicBurner                | 31 |
|--------|-----|---|----|
| Figura | 25. | Programa Tango PCB                                | 32 |
| Figura | 26. | Placa de circuito impresso                        | 32 |
| Figura | 27. | Modelagem do problema                             | 34 |
| Figura | 28. | Mecanismo de detecção do toque                    | 35 |
| Figura | 29. | Detecção do toque                                 | 35 |
| Figura | 30. | Fluxograma do algoritmo de detecção e transmissão | 36 |
| Figura | 31. | Tela Interativa                                   | 37 |
| Figura | 32. | Dimensão da moldura                               | 39 |
| Figura | 33. | Moldura em perspectiva                            | 39 |
| Figura | 34. | Esquema da Placa de Sensores                      | 40 |
| Figura | 35. | Esquema da Placa principal                        | 41 |
| Figura | 36. | Esquema da Placa dos lasers                       | 41 |
| Figura | 37. | Esquema de ligação da tela interativa             | 42 |
| Figura | 38. | Protótipo montado na protoboard                   | 43 |
| Figura | 39. | Fluxograma do algoritmo da placa de sensores      | 44 |
| Figura | 40. | Fluxograma do algoritmo da placa principal        | 45 |

#### LISTA DE SIGLAS

USB - Universal Serial Bus

RAM - Random Access Memory

**EEPROM** - Electrically-Erasable Programmable Read-Only Memory

ALU - Arithmetic Logic Unit

CCS - Custom Computer Services, Inc

ANSI - American National Standards Institute

CP - Code-Protect bit

WRT - Write-Protect bit

WDT - Watchdog Timer

LVP - Low-Voltage Programming

PLL - Phase Locked Loopgsd

PC - Personal computer

Risc - Reduced Instruction Set Computer

PWM - Pulse Width Modulation

USART - Universal Synchronous/Asynchronous

Receiver/Transmitter

IBM - International Business Machines.

ISO - International Organization for Standardization

(Organização Internacional para Padronização)

IEC - International Electrotechnical Commission

NPN - Transistor bipolar com duas camadas de semicondutor tipo

N e uma de tipo P

PNP - Transistor bipolar com duas camadas de semicondutor tipo

P e uma de tipo N

RGB - Red Green Blue

LED - Light Emmiting Diode

CI - Integrated Circuit

CAD - Computer Aided Design

PCI - Peripheral Component Interconnect

HID - Human Interface Device

## LISTA DE TABELAS

| Tabela | 1. | Tipos d | le dados | defini | dos pelo | Padrão | ANSI | C | 17 |
|--------|----|---------|----------|--------|----------|--------|------|---|----|
| Tabela | 2. | Tipos d | de dados | do com | pilador  | PCWH   |      |   | 18 |
| Tabela | 3. | Código  | de core  | s para | resistor | es     |      |   | 24 |

# SUMÁRIO

| 1 INTRODUÇÃO                               | 1  |
|--|----|
| 1.1 OBJETIVOS                              |    |
| 1.2 JUSTIFICATIVAS                         | 2  |
| 1.3 MOTIVAÇÃO                              | 2  |
| 1.4 ESTRUTURA DO TRABALHO                  |    |
| 2 INTERAÇÃO HUMANA NO COMPUTADOR           | 4  |
| 2.1 INTERAÇÕES ATRAVÉS DE CONTATO MANUAL   | 4  |
| 2.1.1 <i>MOUSE</i>                         |    |
| 2.1.2 TECLADO                              | 5  |
| 2.2 INTERAÇÕES ATRAVÉS DE COMANDO DE VOZ   | 6  |
| 2.3 APLICAÇÕES PARA CADA TIPO DE INTERAÇÃO |    |
| 3 COMUNICAÇÃO USB                          | 7  |
| 3.1 CARACTERISTISCAS DA PORTA USB          | 7  |
| 3.2 PADRÃO ELÉTRICO E FISICO DA PORTA USB  | 8  |
| 3.3 PROTOCOLO DE COMUNICAÇÃO USB           | 8  |
| 3.4 FIRMWARE DE UM DISPOSITIVO USB         |    |
| 3.5 APLICAÇÕES USB EM DISPOSITIVOS         |    |
| 4 COMPILADOR C PARA MICROCONTROLADORES     |    |
| 4.1 IDE DE PROGRAMAÇÃO PCWH                |    |
| 4.2 ARQUITETURA DE UM MICROCONTROLADOR PIC |    |
| 4.3 LINGUAGEM C ANSI                       |    |
| 4.4 LINGUAGEM C PADRÃO CCS                 |    |
| 4.5 COMPILANDO UM PROGRAMA EM C            |    |
| 4.6 APLICAÇÕES DE PROGRAMAS EM C           |    |
| 5- COMPONENTES ELETRONICOS E FERRAMENTAS   |    |
| 5.1 COMPONENTES                            |    |
| 5.1.1 RESISTOR                             |    |
| 5.1.2 CAPACITOR                            |    |
| 5.1.3 TRANSISTOR                           |    |
| 5.1.4 DIODO                                |    |
| 5.1.5 DIODO LASER                          |    |
| 5.1.6 RECEPTOR INFRAVERMELHO               |    |
| 5.1.7 CIRCUITO INTEGRADO                   |    |
| 5.2 FERRAMENTAS DE SOFTWARE                |    |
| 5.2.1 PCWH                                 |    |
| 5.2.2 IC-PROG                              |    |
| 5.2.3 PICBURNER                            |    |
| 5.2.4 TANGO PCB                            | 31 |

| 6       | DES          | SENVOLVIM | ENTO          | DA :  | <b>CELA</b>   | INT  | ERATI | VA | <br> | <br>3 | 3 |
|---------|--------------|-----------|---------------|-------|---------------|------|-------|----|------|-------|---|
|         | 6.1          | DEFINIÇÃ  | O DO          | PROI  | BLEMA         | A    |       |    | <br> | <br>3 | 3 |
|         | 6.2          | MODELAGE  | M DO          | PROI  | 3LEM <i>A</i> | A    |       |    | <br> | <br>3 | 3 |
| 7       | IMI          | PLEMENTAÇ | ÃO            |       |               |      |       |    | <br> | <br>3 | 8 |
|         | 7.1          | CONSTRUÇ  | ÃO D <i>i</i> | ES:   | rrutu         | JRA. |       |    | <br> | <br>3 | 8 |
|         | 7.2          | SISTEMA   | ELETF         | RÔNI  | CO            |      |       |    | <br> | <br>3 | 9 |
|         | 7.3          | FIRMWARE  | ·             |       |               |      |       |    | <br> | <br>4 | 3 |
|         | 7.4          | COMUNICA  | ÇÃO (         | COM ( | O COI         | MPUT | ADOR. |    | <br> | <br>4 | 5 |
| 8       | COI          | NCLUSÃO   |               |       |               |      |       |    | <br> | <br>4 | 7 |
| $R^{j}$ | E <i>FE1</i> | RÊNCIA BI | BLIO          | RÁF.  | ICA.          |      |       |    | <br> | <br>4 | 8 |
| A       | PÊNI         | DICE      |               |       |               |      |       |    | <br> | <br>5 | 0 |

### CAPÍTULO 1

## INTRODUÇÃO

Nos dias de hoje é muito comum que uma instituição, seja ela de ensino ou uma empresa, ter como recurso multimídia um projetor de imagens ligado a um computador para palestras, treinamentos, cursos, etc. Analisando desde o início das atividades acadêmicas pude notar em aulas, em que o professor tinha que sair de próximo a tela de apresentação para perto do computador para mudar ou interagir como conteúdo da matéria apresentada, perdendo o tempo e às vezes tirando a linha de raciocínio com o conteúdo.

Com base neste problema, veio à idéia de criar uma forma eficiente e barata de um sistema que interagisse com o conteúdo mostrado pelo computador na tela, de forma a se comportar como uma tela touch screen comumente utilizada em terminais bancários.

Na Alemanha, pesquisadores do Instituto Fraunhofer e da Universidade de Braunschweig estão desenvolvendo telas transparentes multi-toque para projeção. O Khronos Projector é uma instalação interativa onde você volta ou acelera um vídeo no tempo ao tocar na superfície da imagem, num efeito de água, onde o seu toque tem o controle do tempo. Monitores touch-screen já existem a muitos anos, em supermercados, aeroportos, shoppings. O desafio de fazer uma tela multi-touch é que precisamos não só detectar o toque, mas vários toques e gestos que podem, ao mesmo tempo, controlar diferentes ações [1].

#### 1.1 - OBJETIVOS

O objetivo deste projeto visa o desenvolvimento de uma moldura em que o projetor possa imprimir a imagem dentro de sua área, de modo que quando for tocada, a moldura envie para o computador a posição (x, y) do plano cartesiano o local onde foi feito o contado, para que o mesmo faça o tratamento do toque como se fosse um clique de um mouse na posição tocada. A comunicação entre a tela e o computador será feita por meio da porta USB do computador, por ser a interface de comunicação mais difundida nos computadores atualmente.

#### 1.2 - JUSTIFICATIVAS

Com a crescente utilização de sistemas de projeção de imagens por escolas, faculdades, empresas, etc. Fica evidente o tamanho do mercado para um sistema eficiente e prático de interação com o conteúdo apresentado pelo projetor, além de tornar o conteúdo mais dinâmico e eficiente.

Outro fator importante que também justifica o desenvolvimento do projeto é a disposição no mercado de componentes de baixo custo, e que podem ser utilizados no desenvolvimento do projeto, visando criar um sistema barato e de fácil aquisição para instituições onde os recursos financeiros são escassos.

### 1.3 - MOTIVAÇÃO

Uma das motivações do projeto está em mostrar que com componentes comuns no mercado é possível criar um sistema simples e eficiente de interface de interatividade com o conteúdo apresentado pelo projetor, visto que este tipo de equipamento será construído de maneira inédita com o emprego de lasers. A outra motivação do projeto está em desenvolver uma ferramenta que possa de alguma forma ajudar no método de ensino, dando mais liberdade ao apresentador de forma a deixar o conteúdo mais interativo e dinâmico.

#### 1.4 - ESTRUTURA DO TRABALHO

- O presente trabalho está subdividido em nove capítulos que serão descritos a seguir:
- O primeiro capítulo apresenta a contextualização e justificativa para o desenvolvimento da proposta do trabalho;
- O segundo capítulo trata de algumas formas de como o ser humano interage com o computador, e como funcionam alguns periféricos do computador;
- O terceiro capítulo explica superficialmente o padrão USB, pegando desde a parte física até a parte básica do modo de comunicação com o computador, e no final é mostrado um firmware de um dispositivo USB;
- O quarto capítulo apresenta uma breve descrição do compilador utilizado, fazendo uma breve comparação entre a linguagem C ANSI e a linguagem C utilizada para microcontroladores;

- O quinto capítulo apresenta os componentes e as ferramentas utilizadas para o desenvolvimento do projeto.
- O sexto capítulo explica a modelagem do problema, e as formas adotadas para o desenvolvimento da tela interativa.
- O sétimo capítulo apresenta a implementação da montagem e diagrama do sistema da tela interativa, desde a parte eletrônica até a parte da lógica do funcionamento do firmware.
- O oitavo capítulo apresenta as considerações finais, abrangendo as conclusões do desenvolvimento deste trabalho, o conhecimento adquirido no decorrer do desenvolvimento.
- O nono capítulo apresenta a referência bibliográfica utilizada no decorrer do desenvolvimento do projeto.

#### CAPÍTULO 2

## INTERAÇÃO HUMANA NO COMPUTADOR

Neste capítulo será feita uma descrição das formas como o ser humano interage com o computador, desde contatos manuais até comandos de voz. Será visto como cada modo de interação pode ajudar em cada situação sendo que determinadas aplicações ou situações podem necessitar de formas distintas para a interação entre o homem e a máquina.

### 2.1 - INTERAÇÕES ATRAVÉS DE CONTATO MANUAL

As interações mais comuns entre um ser humano e um computador ocorrem por meio de dispositivos como *mouse*, teclado, *tablets*, canetas digitalizadoras e telas *touch screen*. Será feito uma abordagem breve sobre cada dispositivo citado.

#### 2.1.1 - *MOUSE*

O mouse é um dos dispositivos mais utilizados no mundo para a interação entre o homem e o computador, ele consiste em movimentar um cursor na tela por meio da sua movimentação física em uma superfície plana, o movimento físico é detectado por meio de um sensor que pode ser mecânico (encoder) ou ótico (feixe de luz ou laser), como mostra a figura 1. Os movimentos detectados pelo sensor são convertidos em valores computacionais em que seu valor influenciará na movimentação do cursor na tela.

Outro recurso no *mouse* além de movimentar o cursor são as teclas cuja função depende das circunstâncias em que o cursor se encontra na tela.



Figura 1: Mouse. Fonte: [2].

#### 2.1.2 - Teclado

O teclado de computador é um tipo de periférico utilizado pelo usuário para a entrada manual no sistema de dados e comandos. Possuem teclas representando letras, números, símbolos e outras funções, baseadas no modelo de teclado de antigas máquinas de escrever, como mostra a figura 2. Basicamente, os teclados são projetados para a escrita de textos, onde são usadas para esse meio cerca de 50% delas. Além para o controle das funções de um computador e seu sistema operacional. Essas teclas são ligadas a um chip dentro do teclado, onde identifica a tecla pressionada e manda para o PC as informações. O meio de transporte dessas informações entre o teclado e o computador pode ser sem fio (ou Wireless) ou a cabo (PS/2 e USB). O teclado vem se adaptando com a tecnologia e é um dos poucos periféricos que mais se destacam na computação [3].



Figura 2: Teclado. Fonte: [4].

### 2.2 - INTERAÇÕES ATRAVÉS DE COMANDO DE VOZ

Tecnologias de reconhecimento de voz permitem que computadores equipados com microfones interpretem a fala humana, por exemplo, para transcrição ou como método de comando por voz. Tais sistemas podem ser classificados por requererem ou não que o usuário treine o sistema a reconhecer seus padrões particulares de fala, por ter a habilidade de reconhecer fala contínua ou por requerer que o usuário fale pausadamente, e pelo tamanho do vocabulário que é capaz de reconhecer (pequeno, da ordem de dezenas a centenas de palavras, ou grande, com milhares de palavras) [5]. A figura 3 mostra alguns softwares de reconhecimento de voz.



Figura 3: Alguns softwares de reconhecimento de voz. Fonte: [5].

#### 2.3 - APLICAÇÕES PARA CADA TIPO DE INTERAÇÃO

Cada tipo de interação entre o homem e o computador é conveniente ser feito da forma mais fácil e prática possível, pois em determinadas aplicações podem ocorrer debilidades ou dificuldades físicas da parte do usuário ou dos recursos, dificultando a interação do homem com o computador.

Este trabalho tem o objetivo de desenvolver uma interface que possa facilitar a forma de interação com o computador, em que uma moldura será a interface de interação entre o usuário e o computador, assim comportando-se como uma tela sensível ao toque.

#### CAPÍTULO 3

### COMUNICAÇÃO USB

Neste capítulo será feita uma descrição da comunicação USB, apresentando o padrão elétrico USB, as vias de transmissão dados, o protocolo USB e também o firmware de comunicação USB de um dispositivo USB.

#### 3.1 - CARACTERISTICAS DA PORTA USB

O computador atua como um host, ou seja, todos os periféricos ligados no barramento USB tem apenas um mestre controlando, sendo ele o responsável pela requisição de todos os dados trafegados no barramento. Podem ser conectados ao host até 127 dispositivos, diretamente ou através de hubs USB. Cabos individuais USB podem ter até 5 metros; com hubs, os dispositivos podem ficar até 30 metros de distância do host (o equivalente a seis cabos). Com o USB 2.0, o barramento possui uma taxa máxima de transferência de dados de 480 megabits por segundo. Um cabo USB possui dois fios para energia (+5 volts e o fio terra) e um par trançado para a condução dos dados. Nos cabos de energia, o computador poderá fornecer até 500 milliampères de energia a 5 volts.

Os dispositivos de baixa-potência (como o mouse) poderão puxar a energia diretamente do barramento. Os dispositivos de alta-potência (como impressoras) possuem fonte própria de alimentação e exigem mínima energia do barramento. Os hubs podem ter suas próprias fontes de energia para fornecer energia aos dispositivos conectados a ele. Os dispositivos USB são hot-swappable (conectáveis "a quente"), ou seja, podem ser conectados e desconectados a qualquer momento.

Os diversos dispositivos USB podem ser colocados no modo *sleep* (hibernar) pelo computador *host*, quando o computador entrar no modo de economia de energia [6].

#### 3.2 - PADRÃO ELÉTRICO E FÍSICO DA PORTA USB

A comunicação USB ocorre por duas linhas diferenciais chamadas D+ e D-. Um nível "1" é transmitido quando a linha D+ é 0.2V maior que D- e um nível "0" é transmitido quando a linha D+ é 0.2V menor que D-.

Além das linhas D+ e D-, um conector USB possui também duas linhas de alimentação: Vbus que fornece uma tensão de cinco volts nominais para a eventual alimentação do periférico e uma linha de terra (GND) [7].

Os conectores também são padronizados tanto na parte do **host** quanto nos dispositivos, sendo que os conectores do tipo "A" e "Mini-A" são utilizados para o lado do host ou portas *downstream* dos *hubs*, enquanto os conectores "B" e "Mini-B" são utilizados para o lado dos dispositivos ou portas *upstream* dos *hubs*, como mostra a figura 4.

| Tipo   | Imagem da porta | Imagem do conector |
|--------|-----------------|--------------------|
| Tipo A | 4.5mm x 12.0mm  | 1                  |
| Tipo B | 7.3mm x 8.5mm   |                    |
| Mini-A | 3.0mm x 6.8mm   |                    |
| Mini-B | 3.0mm x 6.8mm   |                    |

Figura 4: Conectores USB. Fonte: [8].

## 3.3 - PROTOCOLO DE COMUNICAÇÃO USB

O protocolo USB provê um padrão de troca de mensagens entre um computador (host) e um ou mais dispositivos, a comunicação ocorre após o reconhecimento de um dispositivo no barramento USB que pode ocorrer ao ligar o computador ou quando o dispositivo for conectado no barramento USB. Após a detecção o host interroga todos os dispositivos conectados no barramento e define um endereço para cada um, esse é o processo de

enumeração. Cada dispositivo informa inicialmente o tipo de transferência de dados que irá realizar:

- Interrupção: usado em dispositivos como o mouse ou teclado, onde os dados serão em pequenas quantidades;
- Bulk: usado na transmissão continua de dados em que sua integridade é necessária, usado na transmissão de dados entre unidades de memórias, impressoras e dispositivos onde os dados não podem ser transmitidos com erros;
- Isócrono: usado na transmissão contínua de dados em que a probabilidade de falhas na transmissão seja tolerável, sua capacidade de transmissão é a mais alta no protocolo, pois não possui algoritmo de detecção de erros. São comumente utilizados para transferência de dados de áudio e vídeo.

O host também poderá enviar comandos ou verificar parâmetros por meio de pacotes de controle. Conforme os dispositivos são enumerados, o host manterá um registro da total da largura de banda que todos os dispositivos isócronos e os dispositivos de interrupção exigem. Eles podem consumir até 90% dos 480 Mbps da largura de banda disponível. Após 90% de uso, o host não permite o acesso a qualquer outro dispositivo isócrono ou de interrupção. As transferências de grandes pacotes de dados e de controle utilizam qualquer largura de banda restante (pelo menos 10%).

O USB divide a largura de banda disponível em frames e o host controla os frames. Os frames contêm 1.500 bytes e um novo frame tem início a cada milissegundo. Durante um frame, dispositivos isócronos e de interrupção obterão um slot, assim as larguras de banda necessárias serão garantidas. As transferências de pacotes de dados e de controle utilizam qualquer espaço deixado [6].

#### 3.4 - FIRMWARE DE UM DISPOSITIVO USB

O firmware não passa de um conjunto de regras que rege o funcionamento do sistema. O firmware nada mais é do que um software de um sistema embarcado. O software de um sistema embarcado é semelhante ao de um computador pessoal, podendo ser escrito em diversas linguagens de programação. No Apêndice é apresentado um firmware de um mouse USB da biblioteca do compilador PCWH da CCS.

## 3.5 - APLICAÇÕES USB EM DISPOSITIVOS

Atualmente, a interface USB é a conexão mais utilizada nos dispositivos externos do computador como webcam, mouse, teclados, maquinas fotográficas, memórias e outros. Isto se deve a sua grande facilidade de uso e fácil implementação. Além de dispositivos comerciais, muitas empresas ainda utilizam os padrões de comunicações mais antigos como comunicação serial e paralela, muitas destas empresas tem buscado migrar seus equipamentos para um novo padrão e o USB tem sido a melhor opção do mercado devido a sua aceitação pelo mundo tecnológico.

### CAPÍTULO 4

#### COMPILADOR C PARA MICROCONTROLADORES

Neste capítulo será feita uma descrição do compilador PCWH da empresa CCS, este compilador utiliza a linguagem C como base de linguagem de programação, sendo a linguagem utilizada por ele modificada da linguagem C padrão *ANSI* pelo fato dos microcontroladores possuírem uma arquitetura diferente dos computadores em que o padrão ANSI é obedecido na maioria dos compiladores.

### 4.1 - IDE DE PROGRAMAÇÃO PCWH

O ambiente integrado de desenvolvimento utilizado foi o compilador PCWH da empresa CCS, este compilador foi projetado para os sistemas operacionais *Windows* e suporta toda linha de microcontroladores PIC (séries PIC12, PIC14, PIC16, e PIC18). O IDE na realidade é constituído de três módulos compiladores independentes:

- PCB: para dispositivos de 12bits (series PIC12 e PIC16C5X);
- PCM: para dispositivos de 14bits (series PIC 14000 e PIC16xXXX);
- PCH: para dispositivos de 16bits (serie PIC18).

O usuário pode optar por adquirir os módulos separados ou o pacote completo, dependendo da aplicação em que se destina o compilador [9]. As descrições das principais características do compilador utilizado são:

- Compatibilidade com a padronização *ANSI* e *ISO* (algumas características do compilador não fazem parte do padrão ANSI devido ao fato de serem específicas para a arquitetura PIC);
- Grande eficiência no código gerado;
- Grande diversidade de funções e bibliotecas da linguagem C (padrão ANSI), tais como: entrada/saída serial, manipulação de strings e caracteres, funções matemáticas C, etc.;
- Grande portabilidade de código entre os diversos microcontroladores PIC e inclusive com código escrito para outros microcontroladores ou sistemas. Isto significa que é muito fácil adaptar um programa escrito em C para outro dispositivo ou sistema (seja ele outro PIC, outro tipo de microcontrolador, ou mesmo um programa para PC).

#### 4.2 - ARQUITETURA DE UM MICROCONTROLADOR PIC

Os microcontroladores PIC são classificados pelo seu *opcode*, sendo ele de 12, 14 ou 16 bits. Serão abordadas as características básicas do microcontrolador da serie 16F62X com opcode de 14 bits, foi escolhido este modelo por se tratar de um microcontrolador de fácil compreensão e basicamente ser compatível com os demais microcontroladores PIC, a figura 5 ilustra o diagrama de blocos do microcontrolador pic 16f62x.

# PIC16F62X

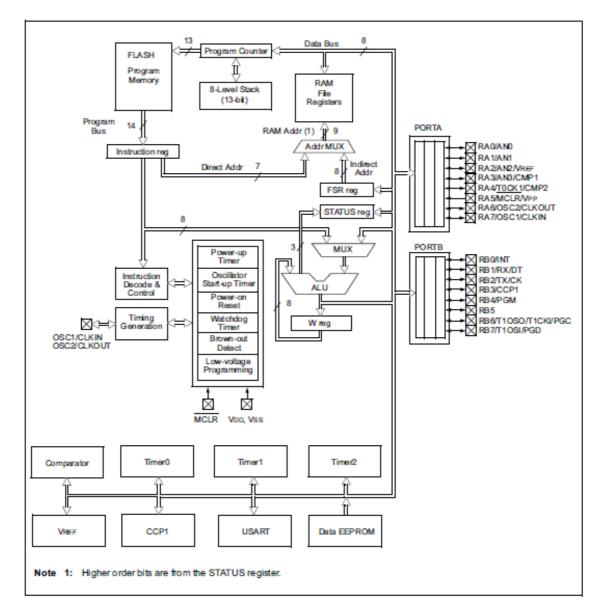


Figura 5: Diagrama de Blocos. Fonte: [10].

As características desse microcontrolador são:

- CPU RISC de alto desempenho:
- Apenas 35 instruções para aprender
- Ciclos de instruções únicos (200ns), exceto os casos de desvios que usam dois ciclos
- Velocidade de operação:
  - DC 20 MHz entrada de clock
  - DC 200 ns ciclo de instrução

- Capacidade para interrupções
- 16 registradores de funções especiais no hardware
- Pilha de oito níveis de profundidade no hardware
- Modo de endereçamento direto, indireto e relativo
- Recursos de periféricos:
- 16 pinos de *I/O* com controle de direção individual
- Alta corrente de saída, podendo acionar leds como carga
- Modulo comparador analógico com:
- Dois comparadores analógicos
- Tensão de referencia programável no chip (VREF)
- Multiplexação programável da entrada do dispositivo e referencia de tensão interna
- Saídas do comparador acessíveis externamente
- Timer0: 8bits timer/counter com prescaler programável de 8bits
- Timer1: 16bits timer/counter com cristal externo/ capacidade de clock
- Timer2: 8bits timer/counter com registrador de período de 8bits, prescaler e postscaler
- Capture, Compare, PWM (CCP) modulo
  - Capture é de 16-bit, max. resolução é de 12.5 ns
  - Compare é de 16-bit, max. resolução é de 200 ns
  - PWM max. resolução é de 10bits
- Universal Sincrono/Assincrono Receptor/
- Transmissor USART/SCI:
- 16 bytes comuns de memória RAM
- Características especiais do microcontrolador:
- Power-on Reset (POR)
- Power-up Timer (PWRT) e tempo de inicialização do Oscilador (OST)
- Detecção de Brown-out (BOD)
- Watchdog (Cão de guarda) Timer (WDT) com próprio oscilador RC confiável
- Pino MCLR multiplexado
- Pull-ups programáveis na PORTB
- Proteção de código programável
- Programável em baixa tensão
- Salva o contexto em modo SLEEP (Descanso)
- Oscilador selecionável

- Bits flash para opções do oscilador
- Oscilador ER (Resistor externo)
- Parte reduzida do count
- Dupla velocidade INTRC
- Pouco consumo de corrente
- EC Entrada de clock externo
- XT Modo de oscilador
- HS Modo de oscilador
- LP Modo de oscilador
- In-circuit Serial Programming™ (via dois pinos)
- Quatro localizações de id de usuários programáveis
- Tecnologia CMOS:
- Pouco consumo, alta velocidade tecnologia CMOS FLASH
- Total concepção estática
- Ampla faixa de tensão de funcionamento
  - PIC16F627 3.0V to 5.5V
  - PIC16F628 3.0V to 5.5V
  - PIC16LF627 2.0V to 5.5V
  - PIC16LF628 2.0V to 5.5V
- Comercial, industrial e larga faixa de temperaturas
- Pouco consumo de energia
  - < 2.0 mA @ 5.0V, 4.0 MHz
  - 15µA típico @ 3.0V, 32 kHz
  - < 1.0μA típica corrente em modo standby @ 3.0V [10]

A *Microchip* fabricante dos microcontroladores PIC, também disponibilizam todo material para a implementação de sistemas de gravação direto na memória de programa de seus microcontroladores, com isso contribuindo para o desenvolvimento de novos softwares como compiladores e interfaces de gravação para os microcontroladores PIC, dessa forma aumentando a flexibilidade dos projetos para estes dispositivos.

A figura 6 apresenta o fluxograma com procedimento de gravação na memória de programa do microcontrolador PIC 16F62x.

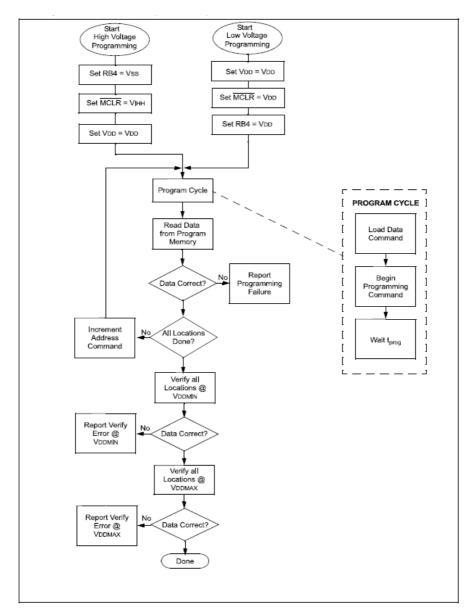


Figura 6: Fluxograma de gravação de dados na memória de programa. Fonte: [11].

#### 4.3 - LINGUAGEM C ANSI

Durante os finais da década de 1970, a linguagem C começou a substituir a linguagem BASIC como a linguagem de programação de microcomputadores mais usada. Durante a década de 1980, foi adaptada para uso no PC IBM, e a sua popularidade começou a aumentar significativamente. Ao mesmo tempo, Bjarne Stroustrup, juntamente com outros nos laboratórios Bell, começaram a trabalhar num projeto onde se adicionavam construções de

linguagens de programação orientada por objetos à linguagem C. A linguagem que eles produziram, chamada C++, é nos dias de hoje a linguagem de programação de aplicações mais comum no sistema operacional *Windows*; C permanece mais popular no mundo *UNIX*.

Em 1983, o instituto norte-americano de padrões (ANSI) formou um comité, X3J11, para estabelecer uma especificação do padrão da linguagem C. Após um processo longo e árduo, o padrão foi completo em 1989 e ratificado como ANSI X3.159-1989 "Programming Language C". Esta versão da linguagem é freqüentemente referida como ANSI C. Em 1990, o padrão ANSI C, após sofrer umas modificações menores, foi adotado pela Organização Internacional de Padrões (ISO) como ISO/IEC 9899:1990, também conhecido como C89 ou C90. Um dos objetivos do processo de padronização ANSI C foi o de produzir um sobreconjunto do K&R C, incorporando muitas das características não-oficiais subseqüentemente introduzidas. Entretanto, muitos programas tinham sido escritos e que não compilavam em certas plataformas, ou com um certo compilador, devido ao uso de bibliotecas de funções não-padrão e ao fato de alguns compiladores não aderirem ao ANSI C [12]. A tabela 1 mostra os tipos de dados adotados no padrão ANSI.

Tabela 1: Tipos de dados definidos pelo Padrão ANSI C. Fonte:[13].

| Tipo               | Tamanho em Bytes | Faixa Mínima                   |
|--------------------|------------------|--------------------------------|
| char               | 1                | -127 a 127                     |
| unsigned char      | 1                | 0 a 255                        |
| signed char        | 1                | -127 a 127                     |
| int                | 4                | -2.147.483.648 a 2.147.483.647 |
| unsigned int       | 4                | 0 a 4.294.967.295              |
| signed int         | 4                | -2.147.483.648 a 2.147.483.647 |
| short int          | 2                | -32.768 a 32.767               |
| unsigned short int | 2                | 0 a 65.535                     |
| signed short int   | 2                | -32.768 a 32.767               |
| long int           | 4                | -2.147.483.648 a 2.147.483.647 |
| signed long int    | 4                | -2.147.483.648 a 2.147.483.647 |
| unsigned long int  | 4                | 0 a 4.294.967.295              |
| float              | 4                | Seis digitos de precisão       |
| double             | 8                | Dez digitos de precisão        |
| long double        | 10               | Dez digitos de precisão        |

## 4.4 - LINGUAGEM C PADRÃO CCS

A linguagem adotada no compilador PCWH foi a linguagem C, porém esta não obedece alguns padrões impostos pela *ANSI* na questão do tamanho das variáveis, visto que microcontroladores são dispositivos com recursos de hardware limitados em relação aos computadores atuais.

A quantidade de memória utilizada para armazenar uma variável depende do seu tipo. A tabela 2 mostra os tipos de dados utilizados no compilador PCWH.

Tabela 2: Tipos de dados do compilador PCWH. Fonte:[9].

| Tipo                   | Tamanho em Bits | Faixa de valores                         |
|------------------------|-----------------|--|
| short int, int1,       | 1               | 0 ou 1                                   |
| boolean                |                 |  |
| char                   | 8               | 0 a 255                                  |
| signed char            | 8               | -128 a 127                               |
| unsigned char          | 8               | 0 a 255                                  |
| int, int8, byte        | 8               | 0 a 255                                  |
| signed int, signed     | 8               | -128 a 127                               |
| byte                   |                 |  |
| unsigned int, unsigned | 8               | 0 a 255                                  |
| byte                   |                 |  |
| long int, int16        | 16              | 0 a 65535                                |
| signed long int        | 16              | -32.768 a 32.767                         |
| unsigned long int      | 16              | 0 a 65535                                |
| int32                  | 32              | 0 a 4.294.967.295                        |
| signed int32           | 32              | -2.147.483.648 a                         |
|                        |                 | 2.147.483.647                            |
| unsigned int32         | 32              | 0 a 4.294.967.295                        |
| float                  | 32              | 1,18 <sup>-38</sup> a 3,4 <sup>+38</sup> |

#### 4.5 - COMPILANDO UM PROGRAMA EM C

Para compilar um programa para microcontrolador primeiro temos que criar um projeto com um arquivo fonte onde serão inseridos os códigos do programa. As figuras 7, 8, 9, 10, 11 e 12 ilustram os passos.

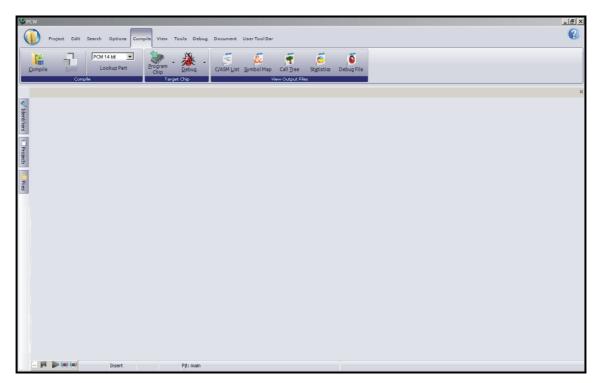


Figura 7: Tela do compilador PCWH.

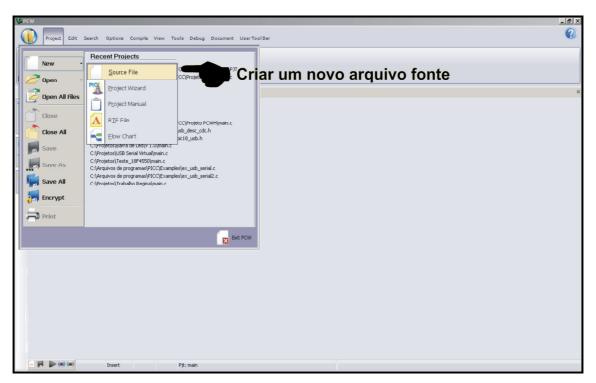


Figura 8: Criando um novo arquivo fonte.



Figura 9: Salvando o arquivo.

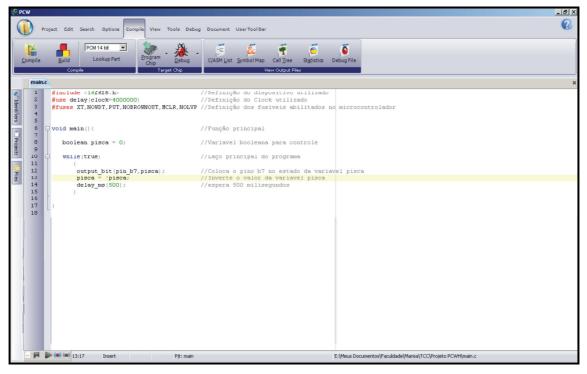


Figura 10: Código de exemplo.

```
Project Cdt Search Options Compile View Tools Debug Document UserToolDay

| Project Cdt Search Options | Indicator Bear | Ind
```

Figura 11: Compilando um arquivo.



Figura 12: Relatório de compilação.

### 4.6 - APLICAÇÕES DE PROGRAMAS C

A linguagem C é recomendada para qualquer tipo de aplicação computacional, sendo que as aplicações mais indicadas nesta linguagem para microcontroladores são aquelas onde a complexidade ou tamanho do projeto são grandes, pois com a utilização de linguagens de baixo nível seria complexo o desenvolvimento desses códigos.

Dentre as linguagens de programação para microcontroladores, a linguagem C se destaca pelo desempenho e por sua facilidade de compreensão, tendo em vista que uns dos fatores que também a fizeram ser bem difundida na área dos microcontroladores foram à extensa lista de bibliotecas, escritas originalmente para computadores.

# CAPÍTULO 5

## COMPONENTES ELETRÔNICOS E FERRAMENTAS

Neste capítulo será feita uma descrição dos componentes eletrônicos utilizados no projeto e das ferramentas utilizadas para o desenvolvimento do projeto como software de simulação de circuitos e de confecção de placas de circuitos impressos.

#### 5.1 - COMPONENTES

Os componentes eletrônicos utilizados em um projeto são de fundamental importância, pois cada um deles tem uma função e especificação própria.

#### 5.1.1 - RESISTOR

Os resistores têm por finalidade apresentar uma resistência elétrica, ou seja, uma oposição à passagem de uma corrente. A medida de resistência é feita numa unidade denominada ohms (W). Os resistores mais comuns são os de carbono e os de fio de níquel cromo ou simplesmente "de fio" e que tem os aspectos mostrados na figura 13.

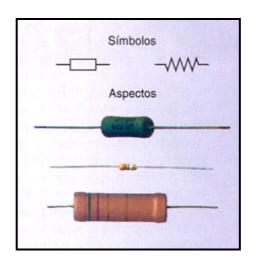


Figura 13: Resistores fixos comuns. (RMF, 2001).

O tamanho do resistor está relacionado com sua capacidade de dissipar o calor. Quanto mais intensa for à corrente num resistor mais calor ele gera e este calor precisa ser transferido ao meio ambiente. É importante

também conhecer o código de cores. As faixas coloridas em torno do resistor dão seu valor conforme mostra a tabela 3. Para ler, o primeiro e o segundo anéis a partir da ponta dão os dois dígitos da resistência enquanto que o terceiro o fator de multiplicação ou número de zeros.

| COR      | 1º anel | 2º anel | 3º anel   | 4º ane |
|----------|---------|---------|-----------|--------|
| Preto    |         | 0       | x 1       |        |
| Marrom   | 1       | 1       | x 10      | 1%     |
| Vermelho | 2       | 2       | x 100     | 2%     |
| Laranja  | 3       | 3       | x 1000    | 3%     |
| Amarelo  | 4       | 4       | x 10000   | 4%     |
| Verde    | 5       | 5       | x 100000  | -      |
| Azul     | 6       | 6       | x 1000000 |        |
| Violeta  | 7       | 7       |           | -      |
| Cinza    | 8       | 8       |           | -      |
| Branco   | 9       | 9       |           | -      |
| Prata    |         |         | x 0,01    | 10%    |
| Dourado  |         |         | v 0 1     | EQ/    |

Tabela 3: Código de cores para resistores. (RMF, 2001).

#### 5.1.2 - CAPACITOR

A finalidade do capacitor é armazenar uma carga elétrica durante um período x. Neste processo o capacitor apresenta algumas propriedades importantes que são aproveitadas em circuitos eletrônicos. Os tipos mais comuns são os cerâmicos, poliéster e eletrolíticos. Os eletrolíticos são polarizados, ou seja, é preciso observar o pólo no momento do uso, como mostra a figura 14. Outra especificação dos capacitores é a tensão máxima que podem suportar ou tensão de trabalho que é medida em volts e que varia entre 3v e 1.200v tipicamente.

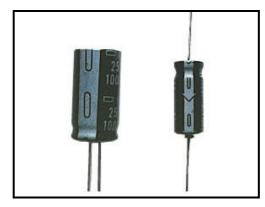


Figura 14: Capacitores: símbolos e aspectos. (RMF, 2001).

## 5.1.3 - TRANSISTOR

Os transistores são componentes formados por três pedaços de materiais semicondutores como o silício P e o silício N formando a estrutura mostrada na figura 15. Nesta figura, também será mostrado o símbolo usado para os dois tipos de transistores mais usados que são os do tipo NPN e PNP.

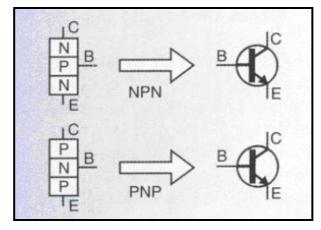


Figura 15: Transistores Bipolares. (RMF, 2001).

Os transistores são os componentes mais importantes dos circuitos eletrônicos, pois podem gerar e amplificar sinais além de funcionar como chaves controladas eletronicamente. A figura 16 mostra os aspectos de alguns transistores comuns. Os transistores de baixa potência que são destinados a trabalhar com correntes pouco intensas fazem parte do chamado grupo (a), enquanto no grupo (b) estão os transistores de média e alta potência que são usados para controlar correntes intensas como, por exemplo, as que circulam por um motor. Estes transistores são dotados de elementos para instalação num radiador de calor, conforme mostra a figura 17. Observe que os transistores possuem terminais de emissor (E), coletor (C) e base (B) e devem ser ligados corretamente em qualquer projeto.

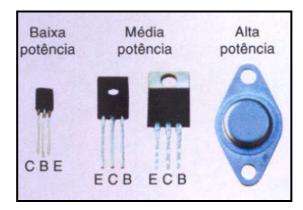


Figura 16: Aspectos dos transistores. (RMF, 2001).

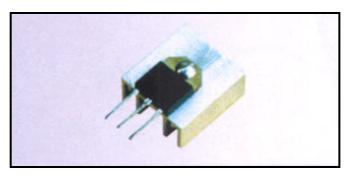


Figura 17: Transistor montado em radiador de calor. (RMF, 2001).

Para usar um transistor é preciso ter em conta a tensão máxima, a corrente máxima e o ganho (fator de amplificação) que pode variar entre 5 e 10.000. Na operação normal, a corrente aplicada à base do transistor controla a corrente que circula entre o emissor e o coletor.

#### 5.1.4 DIODO

Os diodos são componentes semicondutores que conduzem a corrente num único sentido. Na figura 18, são mostrados os símbolos e aspectos dos diodos mais usados nos circuitos práticos de Mecatrônica. Os diodos possuem uma faixa em seu invólucro que permite identificar seu catodo.

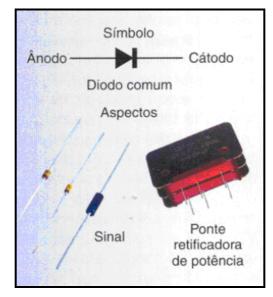


Figura 18: Diodos. (RMF, 2001).

## 5.1.5 - DIODO LASER

Um diodo laser é um pequeno dispositivo semelhante a um diodo emissor de luz (LED). A principal diferença é que um diodo laser produz luz laser coerente enquanto um LED produz luz incoerente. Desde que foi inventado em 1962, milhões de diodos laser tem sido utilizados em uma ampla gama de consumidores e de produtos industriais. A figura 19 ilustra um tipo de diodo laser.



Figura 19: Diodo Laser.

#### 5.1.6 - RECEPTOR INFRAVERMELHO

O receptor infravermelho consiste em um semicondutor sensível a luz infravermelha, convertendo a intensidade da luz aplicada em si em um nível de corrente elétrica por seus contatos. Seu formato se parece com um LED (diodo emissor de luz) como apresentado na figura 20, tem sua aplicação na maioria das vezes em conjunto com um emissor de luz infravermelha.



Figura 20: Receptor Infravermelho.

## 5.1.7 - CIRCUITO INTEGRADO

Após a invenção dos transistores, as implementações de circuitos digitais tornavam-se cada vez mais complexas. Pequenas calculadoras já utilizavam milhares de pequenos transistores. O número desse dispositivo empregado nos circuitos digitais aumentava rapidamente. Com isso veio à necessidade de reduzir a eletrônica envolvida, e desta forma surgiu outra invenção, o circuito integrado (CI), exibido na figura 21 [14].

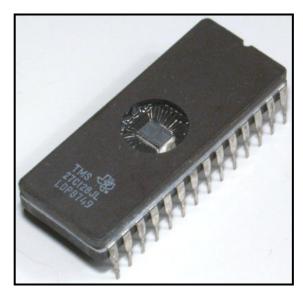


Figura 21: Circuito integrado.

A cada novo CI lançado a complexidade das aplicações aumentava exponencialmente, possibilitando aplicações que antes eram quase impossíveis de implementar. Com o avanço da tecnologia, surgiram os primeiros microprocessadores com uma capacidade maior de execução, isto é, capazes de executar milhões de instruções por segundo. Atualmente, nota se a presença dos circuitos integrados na grande maioria dos equipamentos eletrônicos.

## 5.2 - FERRAMENTAS DE SOFTWARE (SDK)

Outra parte importante para obter sucesso na execução do projeto são os softwares que fazem a simulação dos circuitos.

## 5.2.1 - PCWH

O compilador PCWH é umas das ferramentas principais do projeto, pois será com ele que será desenvolvido todo o firmware da tela interativa, seu ícone para execução é apresentado na figura 22. Após a compilação do projeto o compilador gera um arquivo de extensão '.hex' que conterá todo o programa codificado em hexadecimal, este código é utilizado para ser gravado na memória do microcontrolador.



Figura 22: Ícone do compilador PCWH.

#### 5.2.2 - IC-PROG

O IC-Prog[15] é a ferramenta de software que junto com a interface PICBURNER gravam todo o código gerado pelo compilador na memória do microcontrolador. Primeiramente é definida a interface a ser utilizada no ICPROG, depois é definido o dispositivo (PIC), logo a seguir é carregado o código do firmware do microcontrolador e por ultimo é solicitado a gravação do código na memória do microcontrolador. A figura 23 mostra a tela principal do software de programação IC-Prog.

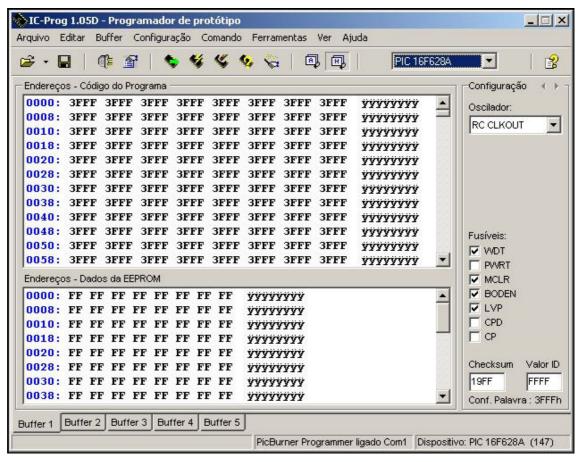


Figura 23: IC-Prog.

#### 5.2.3 - PICBURNER

O PicBuner é uma interface de *hardware*, sendo ela a responsável por transmitir os sinais elétricos diretos ao microcontrolador, sua conexão ocorre através da porta serial do microcomputador, ficando o *software* de gravação de transmitir os códigos já compilados para os endereços da memória interna do microntrolador. Como se pode observar na figura 24, a interface PicBurner é constituída de uma placa com soquete para fixação dos circuitos e um cabo para ligação na porta seria do computador.

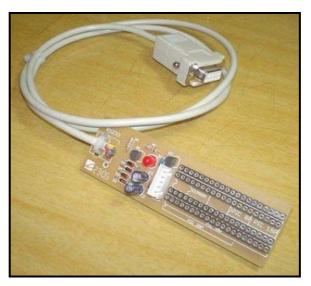


Figura 24: Interface de programação PicBurner.

## 5.2.4 - TANGO PCB

O software tango é uma ferramenta CAD para desenvolvimento de desenhos para placas de circuito impresso (PCI), possui uma extensa biblioteca de componentes elétricos e eletrônicos e também conta com uma interface fácil e amigável para criação de desenhos e componentes eletroeletrônicos. Apesar de ser um software com mais de uma década no mercado, muitas empresas de desenvolvimento eletrônico ainda a utilizam por ser uma ferramenta fácil e eficiente no desenvolvimento de projetos simples. Na figura 25, será apresentado um exemplo de um projeto do desenho de uma placa de circuito em sua área de trabalho, e na figura 26 uma placa de circuito impresso pronta.

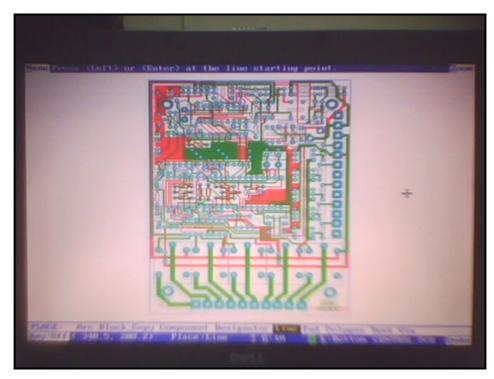


Figura 25: Programa Tango PCB.

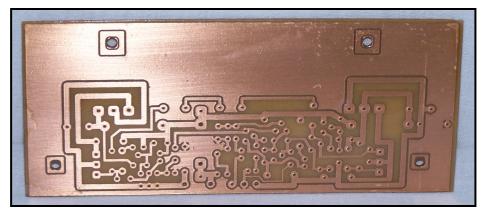


Figura 26: Placa de circuito impresso.

## CAPÍTULO 6

#### DESENVOLVIMENTO DA TELA INTERATIVA

Neste capitulo, será definido o problema a ser tratado neste projeto de pesquisa, além do desenvolvimento de todos os dispositivos eletrônicos necessários. Serão desenvolvidos os softwares responsáveis pela simulação dos circuitos eletrônicos e a confecção da estrutura física da tela interativa.

# 6.1 - DEFINIÇÃO DO PROBLEMA

O problema que será abordado neste projeto de pesquisa, consiste em desenvolver uma tela interativa, onde o usuário ira interagir com o conteúdo da projeção feita na tela, e ocorrerá seguindo o princípio de uma tela touch screen. O sistema será composto por um computador e periféricos (teclado, mouse e opcionalmente caixas de som), um projetor de imagens e a tela interativa. O projetor irá exibir as imagens recebidas através de um cabo rgb ou similar vindo do computador. Estas imagens deverão estar ajustadas perfeitamente dentro da moldura na tela interativa, e a mesma terá que estar conectada no computador por meio de uma conexão USB. O desenvolvimento do projeto será feita em partes e que serão descritas em cada um dos tópicos a seguir.

#### 6.2 - MODELAGEM DO PROBLEMA

A figura 27 mostra a modelagem do problema. Ela foi dividida em módulos para facilitar o desenvolvimento de cada parte que constitui o sistema. Os módulos são: detecção do toque na tela, estrutura física e algoritmo de detecção e transmissão.

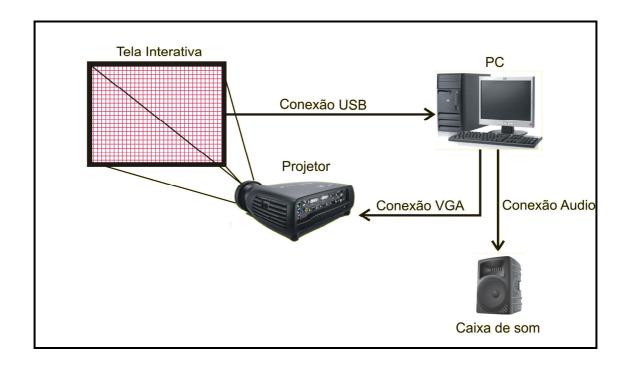


Figura 27: Modelagem do problema.

# MÓDULO 1: DETECÇÃO DO TOQUE NA TELA

A tela irá detectar a presença de toques a partir de lasers e sensores, onde acusará como um toque, a parte onde o contato bloquear a emissão do laser até o sensor. A rotina de detecção do toque será por meio de varreduras feitas em todos os sensores em um determinado tempo, os dados coletados de cada sensor serão utilizados para montar uma matriz, onde a mesma irá representar toda a tela no programa principal do equipamento.

A detecção do toque na tela será feito a partir de sensores receptores de luz infravermelha, quando houver o bloqueio da emissão do laser até o sensor, o sistema irá reconhecer como um toque. A figura 28 ilustra o mecanismo de detecção.

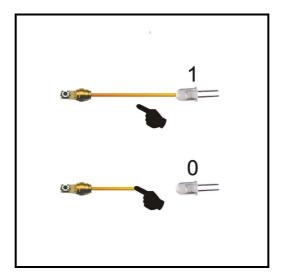


Figura 28: Mecanismo de detecção do toque.

O algoritmo que irá fazer o tratamento dos dados da matriz, será capaz de definir pontos únicos e médios caso mais de um ponto for tocado. Caso possua um feixe desbloqueado entre os dois ou mais tocados, o sistema não irá responder até que se volte à condição normal de funcionamento. A figura 29 demonstra o problema.

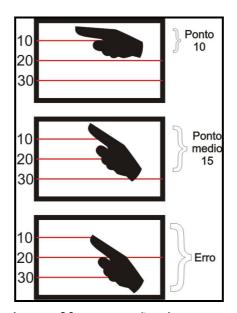


Figura 29: Detecção do toque.

Após o tratamento dos dados da matriz, o sistema terá os dados da posição no plano cartesiano (x, y) referente à área onde foi feito o toque.

Com isso na próxima requisição dos pacotes feito pelo *host*, serão enviados os devidos valores do toque caso tenham ocorrido.

# MÓDULO 2: ALGORITMO DE DETECÇÃO E TRANSMISSÃO

O sistema será constituído de várias placas se comunicando, dentre estas placas uma será a principal, sendo ela a responsável pela comunicação USB e montagem da matriz de estados da tela. O algoritmo será responsável pela varredura constante dos sensores, montagem da matriz e transmissão dos dados pela USB. A figura 30 ilustra a seqüência do programa.

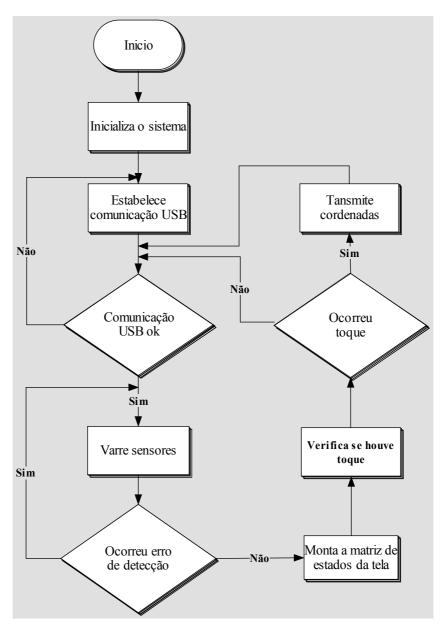


Figura 30: Fluxograma do algoritmo de detecção e transmissão.

## MÓDULO 3: ESTRUTURA FISICA

A estrutura física do sistema será uma moldura onde nas bordas conterão os lasers e os receptores infravermelhos, no centro até as bordas serão de lona especial para projetores de imagens, a estrutura será construída com a característica de um quadro, podendo ser transportada de uma sala para outra. Os feixes de luz do laser não serão visíveis ao olho nu, pois não serão direcionados para quem estiver olhado em sua direção, visto que adornos ou tinturas nas unhas podem desviar os feixes para direção de quem estiver olhando ou para quem estiver entre o feixe desviado. Será notificada a necessidade de uso de luvas para pessoas que possam de alguma forma gerar desvio no feixe de laser.

O bloqueio da imagem do projetor não afetará a reação do toque na tela, pois o mesmo não necessitará da imagem para gerar a coordenada em que será identificado o toque, porém ficaria difícil de o usuário interagir com o conteúdo apresentado caso a projeção não esteja de acordo com as dimensões e visualizações adequadas.

A figura 31 demonstra o projetor enviando imagens para a tela.

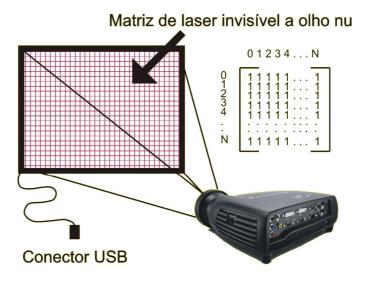


Figura 31: Tela Interativa.

# CAPÍTULO 7

# **IMPLEMENTAÇÃO**

Neste capitulo será detalhado a implementação do projeto, tanto na parte eletrônica quanto na parte de software.

# 7.1 - CONSTRUÇÃO DA ESTRUTURA

A estrutura da tela montada nos testes foi composta por uma moldura de madeira, onde a mesma abriga os lasers, sensores óticos e a parte eletrônica que controla o sistema. As dimensões adotadas para a tela podem ser vistas na figura 32, cada sensor foi fixado a 2,5cm de distância um do outro num total de 40 sensores e lasers na coluna e 48 sensores e lasers na linha.

No fundo da moldura foi fixada uma lona branca, que é muito utilizada para receber a projeção do projetor de imagens, outro fator que ajudou na decisão entre a lona e um fundo feito com outro material, foi o peso total do equipamento, pois com a lona seu peso seria inferior a qualquer outro material visto. Uma melhor idéia da forma física da moldura pode ser vista na figura 33.

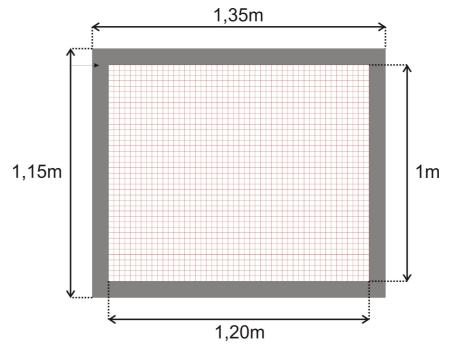
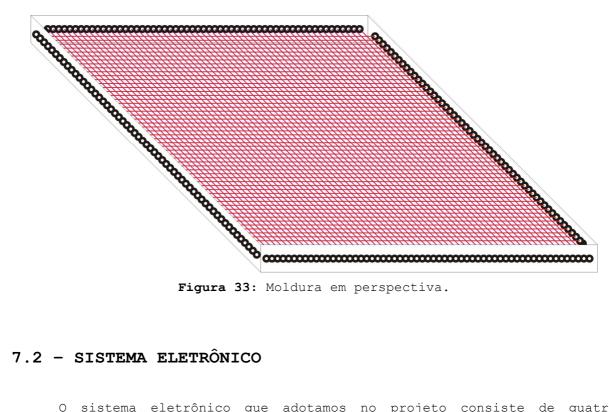


Figura 32: Dimensão da moldura.



O sistema eletrônico que adotamos no projeto consiste de quatro microcontroladores fazendo a coleta dos estados dos sensores óticos e um microcontrolador central recebendo e tratando esses dados, para posteriormente enviá-los ao computador. Na figura 34 vemos o esquema elétrico da placa de sensores.

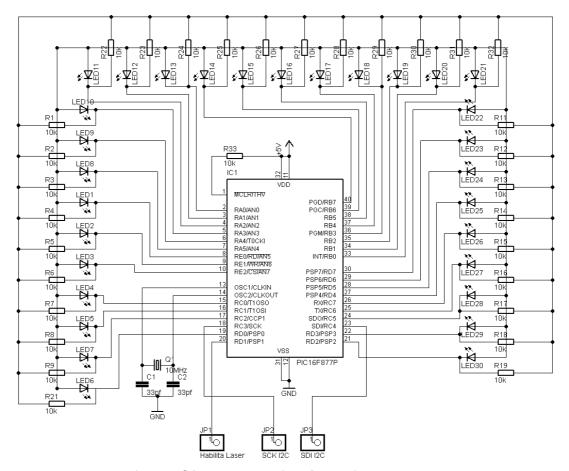


Figura 34: Esquema da Placa de Sensores.

O microcontrolador que receberá os dados de todos os sensores também será encarregado de fazer o tratamento dos dados e envio, este microcontrolador ficará alojado na placa definida como placa principal, a figura 35 mostra o esquema elétrico desta placa.

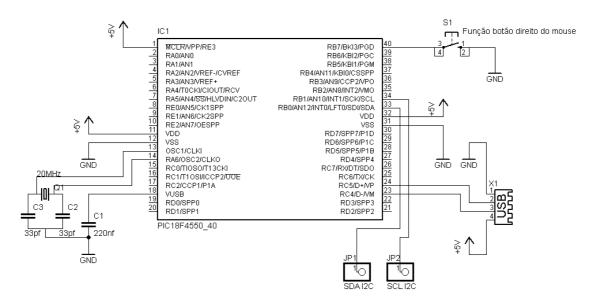


Figura 35: Esquema da Placa principal.

As placas de sensores são conectadas a uma placa contendo os lasers, de forma que, quando é feita a requisição dos estados dos sensores, a placa de sensores envia um sinal para que os lasers pares dos sensores dela sejam energizados. Desta forma apenas os lasers que formam par com os sensores lidos serão ligados, economizando energia por não manter os lasers ligados quando não for necessário. A figura 36 apresenta o esquema elétrico da placa que controla os lasers, e na figura 37 podemos ver a representação do sistema por completo.

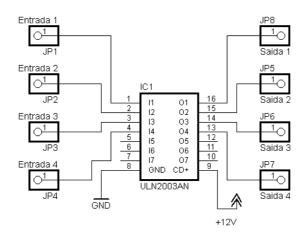


Figura 36: Esquema da Placa dos lasers.

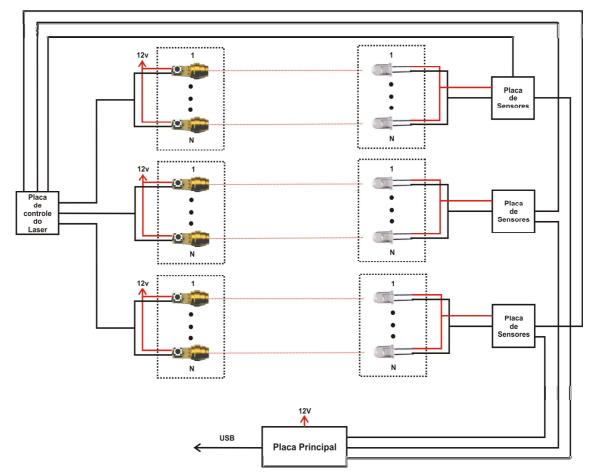


Figura 37: Esquema de ligação da tela interativa.

Para testes preliminares, foi utilizado um *protoboard* (matriz de contato), para montagem do circuito experimental, neste protótipo foi transferido o *firmware* da placa principal, para testes com a comunicação USB. A Figura 38 mostra a montagem do circuito na *protoboard*.

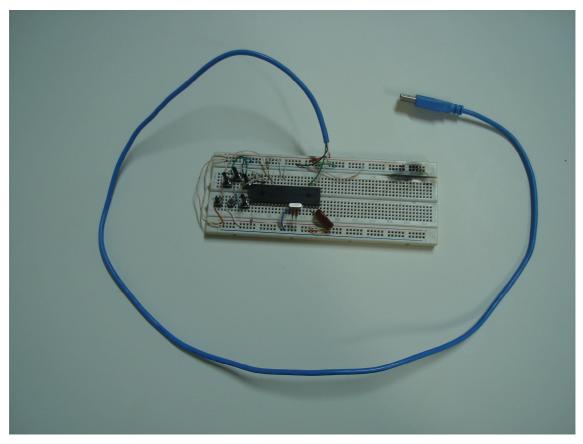


Figura 38: Protótipo montado na protoboard.

### 7.3 - FIRMWARE

Foi desenvolvido em Linguagem C utilizando o compilador CCS [7] o firmware dos microcontroladores utilizados no projeto. E a gravação do firmware na memória do microcontrolador foi feita através da interface Picburner e do software IC Prog [15].

Foram escritos dois firmwares distintos para o sistema, um deles é encarregado pelo tratamento dos toques na tela, este firmware é embutido nos microcontroladores da placa de sensores, sendo que o estado dos sensores somente é passado para a placa principal quando a mesma requisitar estes dados. A figura 39 demonstra o fluxograma do firmware da placa de sensores. Foi atribuído um endereço para cada placa de sensor, de modo que quando a placa principal requisitar a varredura de uma determinada posição ela saiba qual endereço buscar.

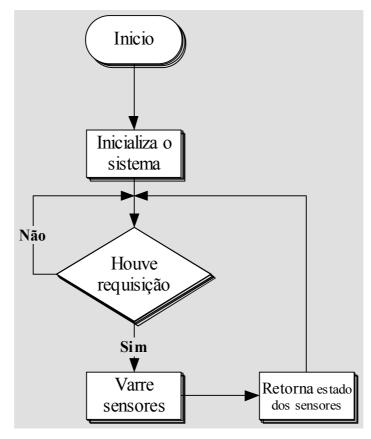


Figura 39: Fluxograma do algoritmo da placa de sensores.

O firmware escrito para a placa principal fica em intervalos determinados coletando os estados dos sensores através do barramento que interliga todos os microcontroladores, quando terminar de receber todos os resultados, a placa principal faz todo o tratamento para detectar se houve um toque, se foi válido e se precisará de algum cálculo de média para centralizar a parte tocada. A figura 40 demonstra o fluxograma do firmware da placa principal.

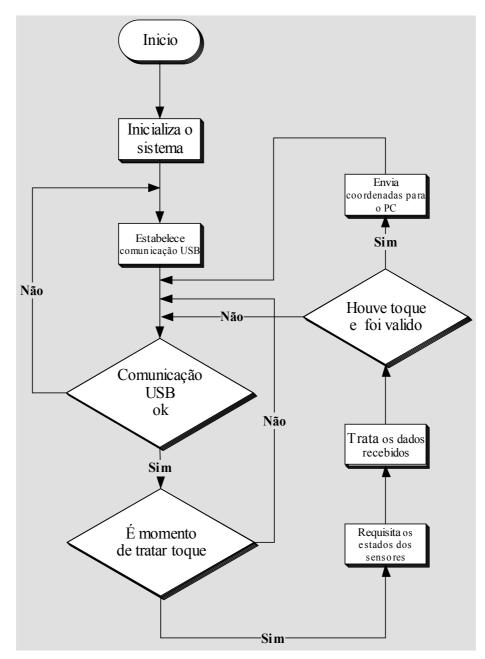


Figura 40: Fluxograma do algoritmo da placa principal.

# 7.4 - COMUNICAÇÃO COM O COMPUTADOR

Por utilizar a classe HID, a tela interativa permite sua utilização em computadores com os sistema operacionais *Windows* 98, ME, XP, Vista e 7. Ao plugar o conector USB na porta do computador, automaticamente o *Windows* carrega um *driver* de *mouse* da classe HID, logo em seguida já habilita a utilização da tela sem que algum *driver* ou software seja necessário. Depois

de carregado o *driver*, a tela estará pronta para ser utilizada desde que a imagem do projetor esteja perfeitamente encaixada dentro da área útil do projetor, caso contrário, as coordenadas do toque não corresponderão com a área vista no ponto do toque.

## CAPÍTULO 8

# CONCLUSÃO

No decorrer deste trabalho foram encontrados vários desafios, uns driblados e outros infelizmente detectados na montagem final do projeto, os maiores enfrentados foram à criação dos descritores para a classe USB, em que os pacotes tiveram que ser escritos para envio da posição x, y do cursor do mouse na tela, sendo que as consultas na pagina que rege o protocolo USB tiveram de ser constantes, para criação dos descritores. A construção física também foi crucial no projeto, pois qualquer movimentação da moldura tirava o ajuste de alguns lasers para a direção dos sensores óticos.

A proposta principal deste trabalho teve como meta demonstrar que com recursos acessíveis é possível desenvolver ferramentas que possam não só ajudar a sociedade de alguma forma, mas também se tornar uma fonte de renda, caso o resultado demonstre viabilidade em se aplicar num produto comercial.

O conhecimento adquirido na realização deste projeto mostrou as possibilidades de se aplicar microcontroladores da família PIC para construção de periféricos para computadores, e abrindo uma nova visão para um segmento pouco explorado pelo mercado brasileiro.

A pesquisa na área de interatividade apesar de ser recente tem contribuído muito com a sociedade nestes anos, pois através disto, a interação com ferramentas, aparelhos e sistemas provocou uma grande evolução nos meios de trabalho, entretenimento, lazer, esporte, educação, etc. Isso mostra que pesquisas na área estão sendo aceitas pela sociedade e por isso tem se mostrado uma boa opção para alunos das áreas de informática.

# REFERÊNCIA BIBLIGRÁFICA

- [1] <a href="http://www.superpink.com/interface/index.php?blogid=1&archive=2006-07">http://www.superpink.com/interface/index.php?blogid=1&archive=2006-07</a> acesso novembro de 2009.
- [2]
   http://www.activewin.com/reviews/hardware/mice/intexplorer3/features.sht
  ml acesso novembro de 2009.
- [3] <a href="http://pt.wikipedia.org/wiki/Teclado">http://pt.wikipedia.org/wiki/Teclado</a> (computador) acesso novembro de 2009.
- [4] <a href="http://www.fnac.com.br/teclado-logitech-deluxe-abnt2-967515--">http://www.fnac.com.br/teclado-logitech-deluxe-abnt2-967515--</a>
  FNAC, informatica-50768-8509.html acesso novembro de 2009.
- [5] <a href="http://informatica.hsw.uol.com.br/reconhecimento-de-voz.htm">http://informatica.hsw.uol.com.br/reconhecimento-de-voz.htm</a> acesso novembro de 2009.
- [6] <a href="http://informatica.hsw.uol.com.br/portas-usb1.htm">http://informatica.hsw.uol.com.br/portas-usb1.htm</a> acesso novembro de 2009.
- [7] Pereira, F.: Tecnologia ARM Microcontroladores de 32 Bits, 1ª. ed. São Paulo , 2007.
- [8] <a href="http://www.intel.com/support/pt/motherboards/desktop/sb/cs-023466.htm">http://www.intel.com/support/pt/motherboards/desktop/sb/cs-023466.htm</a> acesso novembro de 2009.
- [9] Pereira, F.: *Microcontroladores Pic Programação em C*, 6ª. ed. São Paulo , 2005.
- [10] Microchip Technology Inc. PIC16F62X Data Sheet, 2003.
- [11] Microchip Technology Inc. 30034d.pdf, 20032.
- [12] http://pt.wikipedia.org/wiki/C (linguagem\_de\_programa%C3%A7%C3%A3o) acesso novembro de 2009.

- [13] <a href="http://equipe.nce.ufrj.br/adriano/c/apostila/tipos.htm#tipos">http://equipe.nce.ufrj.br/adriano/c/apostila/tipos.htm#tipos</a> acesso novembro de 2009.
- [14] De Oliveira, André Schneider e Souza de Andrade, Fernando.: SISTEMAS

  EMBARCADOS Hardware e Firmware na Prática, 1ª. Ed. São Paulo , 2006.
- [15] Gijnzen, Bonny. Programa IC-Prog versão 1.05D Distribuição Freeware. Disponivel em http://www.ic-prog.com/ e-mail bjijzen@ic-prog.com.

RMF (Revista Mecatrônica Fácil), Eletrônica básica para mecatrônica, Editora Saber, Ano 1,  $N^{\circ}$  1, 2001.

RMF (Revista Mecatrônica Fácil), Futebol de robôs I, Editora Saber, Ano 1,  $N^{\circ}$  4, 2002A.

RMF (Revista Mecatrônica Fácil), Futebol de robôs II, Editora Saber, Ano 1,  $N^{\circ}$  5, 2002B.

RMF (Revista Mecatrônica Fácil), Os robôs que jogam futebol III, Editora Saber, Ano 1,  $N^{\circ}$  6, 2002C.

RMF (Revista Mecatrônica Fácil), Construa um robô controlado pelo PC, Editora Saber, Ano 1,  $N^{\circ}$  8, 2002D.

# **APÊNDICES**

## Arquivo ex usb mouse.c

```
////
                                                                ////
                           ex usb mouse.c
////
                                                                   ////
//// An example of creating a USB mouse. Uses your operating
                                                                ////
//// system's HID drivers, which on most systems should already be ////
//// installed.
                                                               ////
////
                                                                  ////
//// If everything is working, the mouse cursor will move in a
                                                                ////
//// circle when connected to a PC.
                                                                ////
////
                                                                  ////
//// For a more generic USB HID example (as well as some HID
                                                                ////
//// documentation) see ex usb hid.c
                                                                ////
////
                                                                  ////
/// NOTE ABOUT ENDPOINT BUFFER SIZE:
                                                                ////
////
                                                                  ////
//// Although this application sends 4 bytes to the PC, this demo ///
//// defines USB EP1 TX SIZE to 8 to allocate 8 bytes for this
                                                                ////
/// endpoint. This constant is also used in the endpoint
                                                                ////
/// descriptor to specify the endpoint max size. If you were
                                                                ////
//// concious of RAM savings you could redefine these to 4
                                                                ////
//// (or even 1!), but you would lose throughput. The reason for
                                                                ////
//// throughput loss is that if you send a packet that is the same ///
/// size as the max packet size then you need to send a 0 len
                                                                ////
//// packet to specify end of message marker. The routines
                                                                ////
//// usb puts() and usb gets() send and receive multiple packet
//// message, waiting for a 0 len packet or a packet that is smaller \ensuremath{///}
//// than max-packet size.
                                                                ////
////
                                                                   ////
////
//// VERSION HISTORY
                                                                ////
////
                                                                   ////
//// June 20th, 2005: 18Fxx5x Release
                                                                ////
                                                                   ////
//// March 21st, 2005: Initial PIC18Fxx5x support
                                                                ////
```

```
////
                                                               ////
//// June 24th, 2004: Cleanup to work with updated USB API.
                                                            ////
////
                    Mouse now goes in a circle.
                                                            ////
////
                                                               ////
(C) Copyright 1996,2005 Custom Computer Services
//// This source code may only be used by licensed users of the CCS///
//// C compiler. This source code may only be distributed to other///
//// licensed users of the CCS C compiler. No other use,
                                                            ////
//// reproduction or distribution is permitted without written
                                                            ////
//// permission. Derivative programs created using this software ////
/// in object code form are not restricted in any way.
                                                            ////
//set to 1 to use a PIC with an internal USB Peripheral
//set to 0 to use a National USBN960x peripheral
#define USB PIC PERIF 1
#if USB PIC PERIF //use the PIC16C7x5 peripheral
 #define LED1 PIN A5
 #if defined( PCM )
 #include <16C765.h>
 #device *=16
 #fuses HS, NOWDT, NOPROTECT
 #use delay(clock=24000000)
 #else
 #include <18F4550.h>
 //~~~ 20MHZ OSCILLATOR CONFIGS ~~~//
 //// FULL SPEED
  #fuses HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL5, CPUDIV1, VREGEN
  #use delay(clock=48000000)
 ///SLOW SPEED
 //#fuses HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL5, CPUDIV3, VREGEN
 //#use delay(clock=24000000)
 //#define USB USE FULL SPEED FALSE
 #endif
#else //use the National USBN960x peripheral
 #define LED1 PIN B3
```

```
#if defined( PCM )
 #include <16F877A.h>
 #device *=16
 #fuses HS, NOWDT, NOPROTECT, NOLVP
 #elif defined( PCH )
 #include <18F452.h>
 #fuses HS, NOPROTECT, NOLVP, NOWDT
#endif
#use delay(clock=20000000)
#endif //endif check to see which peripheral to use
#use rs232(baud=9600, xmit=PIN C6, rcv=PIN C7)
//
//
// CCS Library dynamic defines. For dynamic configuration of the CCS
Library
// for your application several defines need to be made. See the comments
// at usb.h for more information
//
#DEFINE USB HID DEVICE TRUE //Tells the CCS PIC USB firmware
                        //to include HID handling code.
//turn on EP1 for IN interrupt transfers. (IN = PIC -> PC)
#define USB EP1 TX ENABLE USB ENABLE INTERRUPT
#define USB EP1 TX SIZE 8  //max packet size of this endpoint
//
// If you are using a USB connection sense pin, define it here. If you are
// not using connection sense, comment out this line. Without connection
// sense you will not know if the device gets disconnected.
//
      (connection sense should look like this:
//
                        100k
//
          VBUS-----(I/O PIN ON PIC)
```

```
//
//
               +----GND
//
                    100k
//
      (where VBUS is pin1 of the USB connector)
///only the 18F4550 development kit has this pin
#if USB PIC PERIF && defined( PCH )
#define USB CON SENSE PIN PIN B2
#endif
//
//
// Include the CCS USB Libraries. See the comments at the top of these
// files for more information
#if USB PIC PERIF
#if defined( PCM )
 #include <pic usb.h> //Microchip PIC16C765 hardware layer for usb.c
#else
 #include <pic18 usb.h> //Microchip PIC18Fxx5x hardware layer for usb.c
#endif
#else
#include <usbn960x.c> //National 960x hardware layer for usb.c
#endif
#include <usb desc mouse.h> //USB Configuration and Device descriptors
for this UBS device
#include <usb.c>
            //handles usb setup tokens and get descriptor
reports
// Configure the demonstration I/O
//
#define LED2 PIN B4
```

```
#define LED3 PIN B5
#DEFINE BUTTON PIN A4
#define LED ON output low
#define LED OFF output high
// usb debug task()
//
// When called periodically, displays debugging information over serial
// to display enumeration and connection states. Also lights LED2 and LED3
// based upon enumeration and connection status.
//
void usb debug task(void) {
  static int8 last connected;
  static int8 last enumerated;
  int8 new connected;
  int8 new enumerated;
  new connected=usb attached();
  new enumerated=usb enumerated();
  if (new_connected)
     LED ON(LED2);
  else
     LED OFF (LED2);
  if (new enumerated)
     LED ON(LED3);
  else
     LED OFF (LED3);
  if (new connected && !last connected)
     printf("\r\n\nUSB connected, waiting for enumaration...");
  if (!new_connected && last_connected)
     printf("\r\n\nUSB disconnected, waiting for connection...");
  if (new_enumerated && !last_enumerated)
     printf("\r\n\nUSB enumerated by PC/HOST");
  if (!new enumerated && last enumerated)
```

```
printf("\r\n\nUSB unenumerated by PC/HOST, waiting for
enumeration...");
  last connected=new connected;
  last enumerated=new enumerated;
}
void main(void) {
   #define MOUSE SEQUENCE STEPS 16
  const char mouse_seq[MOUSE_SEQUENCE_STEPS] =
      \{0, 1, 3, 4, 4, 4, 3, 1, 0, -1, -3, -4, -4, -4, -3, -1\};
  int8 out data[4];
  int8 x seq=0; int8 y seq=MOUSE SEQUENCE STEPS/4;
  int8 count=0;
  LED_ON(LED1);
  LED OFF (LED2);
  LED OFF (LED3);
  printf("\r\n\nCCS USB Mouse Example");
  #ifdef PCH
  printf("\r\nPCH: v");
  printf( PCH );
  #else
  printf("\r\n\PCM: v");
  printf( PCM );
  #endif
  usb init cs();
  #if !(__USB_PIC_PERIF__)
  printf("\r\nUSBN: 0x%X", usbn_get_version());
  #endif
  printf("\r\n");
  while (TRUE) {
     usb task();
     usb debug task();
```

```
if (usb_enumerated()) {
         out_data[0]=0; //button state goes here
        out_data[1]=mouse_seq[x_seq];
         out data[2]=mouse seq[y seq];
         out data[3]=0; //wheel state goes here
         if (usb_put_packet(1,out_data,4,USB_DTS_TOGGLE))
            count++;
         if (count > 10) {
            if (++x seq>=MOUSE SEQUENCE STEPS) {x seq=0;}
            if (++y_seq>=MOUSE_SEQUENCE_STEPS) {y_seq=0;}
            count=0;
         }
        delay_ms(10);
      }
  }
}
```