

Centro Universitário Positivo - UNICENP
Núcleo de Ciências Exatas e Tecnológicas – NCET
Engenharia da Computação
Belmiro Sebastião Marques Junior

Geladeira Inteligente

Curitiba
2006

Centro Universitário Positivo - UnicenP
Núcleo de Ciências Exatas e Tecnológicas – NCET
Engenharia da Computação
Belmiro Sebastião Marques Junior

Geladeira Inteligente

Monografia apresentada à disciplina de Projeto Final, como requisito parcial à conclusão do Curso de Engenharia da Computação. Orientador: Prof. Marcelo Mikoze Gonçalves

Curitiba
2006

TERMO DE APROVAÇÃO

Belmiro Sebastião Marques Junior

Geladeira Inteligente

Monografia aprovada como requisito parcial à conclusão do curso de Engenharia da Computação do Centro Universitário Positivo, pela seguinte banca examinadora:

Prof. Marcelo Mikoze Gonçalves (Orientador)

Prof. Maurício Perreto

Prof. Adriana Cursino Thomé

Curitiba, 11 de Dezembro de 2006.

SUMÁRIO

LISTA DE FIGURAS	i
LISTA DE TABELAS	ii
LISTA DE SIGLAS	iii
RESUMO	iv
ABSTRACT	v
1. INTRODUÇÃO	11
1.1 Motivação	11
1.2 Definição do Trabalho	12
1.3 Contextualização nos dias atuais	12
1.4 Descrição das Principais Funcionalidades	12
1.4.1 Ler Códigos de Barras	12
1.4.2 Comunicação Ethernet	13
1.4.3 Acesso aos Dados	13
1.5 Tecnologias Utilizadas na Implementação	13
2. FUNDAMENTAÇÃO TEÓRICA	14
2.1 Teoria Pura e Aplicada a Prática	14
2.1.1 Códigos de Barras	14
2.1.2 Tipo de Códigos de Barras	14
2.1.3 Códigos de Barras mais Comuns	15
2.2 Teoria de Software	16
2.2.1 Linguagem C	16
2.3 Teoria de Hardware	18
2.3.1 Microcontrolador	18
2.3.1.1 Microcontrolador Rabbit	18
2.3.2 Leitor de Códigos de Barras	20
2.3.3 Display LCD	21
2.3.4 Ethernet	23
3. ESPECIFICAÇÃO TÉCNICA	25
3.1 Especificação de Hardware	25
3.2 Especificação de Software	26
3.3 Aplicabilidades/Funcionalidades de Hardware	26
3.4 Aplicabilidades/Funcionalidades de Software	28

4. PROJETO	29
4.1 Projeto de Hardware	29
4.1.1 Display LCD	29
4.1.2 Leitor de Códigos de Barras	30
4.1.3 Protocolo de Comunicação	31
4.1.4 Kit do Microcontrolador Rabbit	32
4.2 Projeto de Software	33
4.2.1 Diagrama de Caso de Uso	33
4.2.2 Descrição de Caso de Uso	34
4.2.3 Diagrama de Seqüência	35
4.2.4 Estrutura para Armazenamento de Dados	36
5. CRONOGRAMA DO PROJETO	37
6. VALDAÇÃO	38
7. ESTUDO DE VIABILIDADE- ECONÔMICA	38
8. RESULTADOS	40
9. CONCLUSÃO	44
10. REFERÊNCIAS BIBLIOGRÁFICAS	45

LISTA DE FIGURAS

Figura 1 - Microcontrolador Rabbit RCM3000	19
Figura 2 – Kit de desenvolvimento Rabbit	20
Figura 3 – Display LCD 16 x 2	23
Figura 4 – Rede Ethernet	24
Figura 5 – Diagrama de blocos do hardware	26
Figura 6 – Diagrama de blocos do software.....	27
Figura 7 – Funcionalidade do hardware do sistema.....	32
Figura 8 – Esquemático de interligação entre Rabbit e display LCD.	30
Figura 9 - Esquemático de interligação entre Rabbit e o leitor de código de barras.	33
Figura 10 – Pinagens do Rabbit (conectores J1 e J2).....	34
Figura 11 - Diagrama de caso de uso	33
Figura 12– Diagrama de seqüência visualizar dados	35
Figura 13– Diagrama de seqüência adicionar dados	35
Figura 14– Diagrama de seqüência remover dados	36
Figura 15 - Projeto internamente com o kit do Rabbit	40
Figura 16 - Projeto externamente com as conexões	41
Figura 17 - Projeto externamente Montado	42
Figura 18 – Página da Web do projeto.....	43

LISTA DE TABELAS

Tabela 1 - LCD linhas x Colunas x Pinos	21
Tabela 2 – Descrições e funções dos pinos	22
Tabela 3 - Descrição dos sinais do display LCD	29
Tabela 4 - Descrição dos sinais do leitor de código de barras	30
Tabela 5 - Estrutura de dados	36
Tabela 6 – Cronograma do projeto	37
Tabela 7 – Estimativa de custo do Projeto	38

LISTA DE SIGLAS

CGI - Common Gateway Interface

IP – Internet Protocol

NCET - Núcleo de Ciências Exatas e Tecnológicas

RAM – Random Access Memory

ROM – Read Only Memory

SGBD – Sistema Gerenciador de Banco de Dados

SQL – Structured Query Language

TCP – Transmission Control Protocol

UART – Universal Asynchronous Receiver/Transmitter

UNICENP – Centro Universitário Positivo

RESUMO

Este projeto tem por objetivo apresentar um sistema de controle de alimentos de uma geladeira com acesso dos dados através de uma pagina web rodando de dentro de um microcontrolador.

É desenvolvido no microprocessador Rabbit da Zworld, com uma entrada e saída de dados sendo feita através de um leitor código de barras e podendo ser visualizados através de um display LCD.

Este projeto tem por objetivo também facilitar a vida das pessoas que não dispõem de muito tempo para isso, dessa forma sendo facilitada uma vez que pode acessar esses dados através da Internet.

Palavras-chave: microprocessador rabbit, comunicação Internet, leitor de código de barras.

ABSTRACT

This Project has the purpose to present a food control system of a refrigerator that its datas could be access by the web page running inside of the microcontroller.

It is developed in the microprocessor Rabbit from Zworld, where the input e output data is made by the barcode reader e the data colud be Access by the display LCD.

This project has the purpose also to facilitate the life for people who don't have enough time to do it and in this way this data can be Access by the Internet that is very good easy way to do it.

Keywords: microprocessor rabbit, Internet communication, barcode reader.

1. INTRODUÇÃO

Atualmente a necessidade de facilitar a vida de pessoas que vivem sozinhas e também não disponham de tempo para ir ao supermercado e mesmo fazer um controle dos alimentos que dispõem em sua geladeira.

A idéia da geladeira inteligente é criar um sistema de controle de entrada e saída de alimento na geladeira, de modo que possa ser acessada através de uma página na Internet a lista de alimentos contidos nela.

Se uma empresa pudesse ter acesso à lista de alimentos dos clientes e fornecesse a compra e a entrega, poderia direcionar suas campanhas publicitárias.

1.1. MOTIVAÇÃO

É recompensador chegar ao último ano do curso de engenharia da computação, poder aplicar os conhecimentos adquiridos ao longo do curso e perceber o quanto enriqueceu em quantidade de conhecimento.

E ao trabalhar com novas tecnologias vem a provar o quanto é amplo a área tecnológica e também o quanto é rápido a evolução nessa área, portanto ao me deparar com isso pude perceber que na faculdade não aprendemos só o que nos é ensinado, mas na verdade o modo como aprender.

Além disso, trabalhar com o microprocessador Rabbit RCM 3000 vai ser importante uma vez que não é muito utilizado na faculdade e não conheço seu funcionamento.

Um dos motivos pelo qual o tema desse projeto foi escolhido se deve ao fato do grande interesse a área de automação comercial e residencial.

1.2. DEFINIÇÃO DO TRABALHO

O projeto proporciona ao usuário controlar os alimentos de sua geladeira podendo visualizar através de uma página de Internet que ficará rodando a partir de um microcontrolador.

Dessa forma, o projeto é formado por um microcontrolador que controla todo o sistema, os periféricos, sendo comunicado via Ethernet com o computador.

1.3. CONTEXTUALIZAÇÃO NOS DIAS ATUAIS

Nos dias atuais é notado que a tecnologia vem sendo desenvolvido em função do homem, a fim de facilitar sua vida, e também automatizando fábricas, indústrias que tem seus processos de trabalhos sendo substituídos.

Assim sendo, a geladeira inteligente se encaixa nesse novo seguimento de automatização da tecnologia, tendo em vista que se deseja facilitar a vida do homem, uma vez que se propõe a mostrar através de uma página de Web, a lista de alimentos contidos na sua geladeira, enfim um controle total sobre os alimentos inseridos e retirados da geladeira.

1.4. DESCRIÇÃO DAS PRINCIPAIS FUNCIONALIDADES

As principais funcionalidades do sistema serão apresentadas

1.4.1. Ler Códigos de Barras

O sistema lê e interpreta os códigos de barras dos produtos alimentícios e armazena ou remove da memória.

1.4.2. Comunicação Ethernet

O sistema é capaz de comunicar pc com o microcontrolador via Ethernet.

1.4.3. Acesso aos Dados

O sistema utiliza acesso a um mini banco de dados rodando dentro do microcontrolador Rabbit, podendo incluir, registrar ou remover informações.

1.5. TECNOLOGIAS UTILIZADAS NA IMPLEMENTAÇÃO

- Microprocessadores Rabbit
- Comunicação Ethernet

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada a teoria das tecnologias a serem utilizadas para o desenvolvimento do projeto.

2.1. TEORIA PURA E APLICADA A PRÁTICA

2.1.1. CÓDIGOS DE BARRAS

Seqüência de barras pretas e brancas, indecifráveis para nós, nada mais é que a representação de um pequeno conjunto de números e/ou letras, impressos de uma forma que o leitor óptico possa interpretar: o preto retém a luz e o branco a reflete, de forma que o leitor capture os sinais e interprete qual a seqüência de números (ou letras) representada pelas barras [3].

2.1.2. TIPOS DE CÓDIGOS DE BARRAS

Há diversos tipos de códigos, portanto os desenhos das barras variam, ou seja, diferentes critérios para combinar barras claras e escuras. A seqüência “123”, por exemplo, pode ter diversas representações, dependendo do tipo de código utilizado.

Portanto, um leitor óptico não é capaz de ler qualquer código de barras, é necessário que esteja devidamente habilitado (configurado) para cada tipo que lhe for apresentado, de modo a conseguir interpretar o código. Os leitores a laser (muito usados no comércio) geralmente são configurados por comandos de programação impressos em menus (cartões) de códigos de barras, ou então por envio de comandos pela porta serial do micro. A grande maioria das lojas de varejo tem seus equipamentos configurados para ler os padrões EAN e UPC, mas geralmente também lêem o código 3 de 9.

A principal vantagem de se usar o padrão internacional é que cada produto terá seu código exclusivo, aplicável no mundo inteiro, sem repetição, o que possibilita a integração e a troca de informações entre os vários elos da cadeia produtiva: do fabricante ao consumidor final, o código acompanha o produto.

2.1.3. CÓDIGOS MAIS COMUNS COMERCIALMENTE

EAN13 é o código mais usado na identificação de itens comerciais. É composto de 13 dígitos: os 3 primeiros representam o país (o Brasil é 789), os 4 seguintes representam o código da empresa filiada à EAN, os próximos 5 representam o código do item comercial dentro da empresa, e o 13º dígito é o verificador, obtido por meio de cálculo algoritmo. De acordo com a grade de itens da empresa (quantidade), a composição pode ser mudada para que o item comercial tenha de 3 a 6 dígitos, e a empresa tenha 6 a 3. Ou seja, a combinação de código da empresa + código do item deve ter 9 dígitos[3].

EAN8 é a versão reduzida do EAN13, para embalagens pequenas. Ele não inclui o código da empresa, apenas o do país (3 dígitos), do produto (4 dígitos) e dígito verificador. Como não há o código da empresa, o licenciamento de numerações é integralmente controlado pela EAN, mediante comprovação técnica de necessidade, taxa de licenciamento e anuidade de manutenção, pois há poucos códigos disponíveis por país[3].

UPC-A tem a mesma aplicação do EAN13, mas é usado para itens comercializados nos Estados Unidos e Canadá. Tem 1 dígito que representa a categoria do produto, 5 dígitos para a identificação do fabricante, 5 dígitos para a identificação do produto e 1 dígito verificador (totalizando 12). Não há identificação do país, pois o cadastro de filiados é único para EUA e Canadá.

UPC-E é a versão UPC de 8 dígitos, obtida suprimindo 4 zeros do UPC-E (entre número da empresa e número do item)[3].

3 de 9 é um código simples de ser gerado, aceita letras e números e é livre. Sua simbologia não é densa, dispensando o uso de dígito verificador (que é usado nos outros padrões para garantir que a leitura foi feita corretamente). Geralmente é encontrado nas em indústrias, companhias aéreas, que necessitam de caracteres alfas-numéricos [3].

Normalmente, os sistemas no Brasil estão modelados para receber código de 13 posições (embora a EAN recomende 14), o que possibilita trabalhar também com códigos de 8 (EAN8/UPC-E) e 12 (UPC-A) posições.

2.2. TEORIA DE SOFTWARE

2.2.1. LINGUAGEM C

O C é uma linguagem de programação genérica que é utilizada para a criação de programas diversos como processadores de texto, planilhas eletrônicas, sistemas operacionais, programas de comunicação, programas para a automação industrial, gerenciadores de bancos de dados, programas de projeto assistido por computador, programas para a solução de problemas da Engenharia, Física, Química e outras Ciências, etc. Apresenta características das linguagens de alto e de baixo nível.

O American National Standards Institute (ANSI) definiu uma versão padrão da linguagem, conhecida como ANSI C. É interessante que os programas sejam desenvolvidos neste padrão para aumentar a sua portabilidade.

C é uma linguagem compilada, isto é, o código fonte é inteiramente transformado para um código executável antes de ser executado. Na verdade o desenvolvimento de um programa em C apresenta 4 fases distintas: edição, compilação, ligação (linking) e execução[4].

Dynamic C é um sistema de desenvolvimento integrado para o software embarcado. É projetado para ser usado com os microcontroladores *Rabbit* e os outros controladores baseados no microprocessador *Rabbit*.

Dynamic C integra as seguintes funções do desenvolvimento:

- Editando
- Compilando
- Referenciando (*Linking*)
- Carregando
- Depurando

Os programas do *Dynamic C* podem ser executados e rodado passo a passo (*debugged*) interativamente no código-fonte. Os menus e os atalhos do teclado para a maioria dos comandos fazem do *Dynamic C* uma ferramenta fácil de se usar.

Dynamic C também suporta programação em Linguagem *assembly*. Não é necessário deixar de usar a linguagem C, para escrever o código na linguagem *Assembly*. As Linguagens C e *assembly* podem ser usados no mesmo código.

O Debugging sob *Dynamic C* inclui a habilidade de usar comandos do `printf`, expressões *watch* e *breakpoints*. As expressões *watch* podem ser usadas para computar as expressões de C que envolvem as variáveis ou as funções. As expressões *watch* podem ser avaliadas quando paradas em um ponto determinado (*breakpoint*).

Dynamic C 9 introduz características de depuração avançadas tais como a execução e o traçado da pilha. O traçado de execução pode ser usado seguindo a execução de indicações depuráveis, incluindo informação como a função/nome do arquivo, números da linha de código da fonte e da coluna, ação executada, índices do registro. O traçado da pilha mostra seqüências da ligação de controle e valores de parâmetro.

Dynamic C fornece extensões a linguagem C (tal como variáveis, *costatements* e *cofunctions* compartilhados e protegidos) esse desenvolvimento encaixado *real-world* do sistema da sustentação.

Dynamic C vem com muitas bibliotecas da função, tudo no código de fonte.

Dynamic C compila diretamente à memória. As funções e as bibliotecas são compiladas e ligadas e *downloaded*. Em um PC rápido, *Dynamic C* pôde carregar 30.000 bytes do código em 5 segundos em uma taxa de baud de 115.200 bps.

Dynamic C difere de um sistema de programação tradicional de C que funciona em um PC ou sob o UNIX. A razão? Para ajudar os clientes a escrever o software de controle embarcado com a maior confiança possível. Não é possível usar C padrão em um ambiente embarcado sem fazer adaptações. A Linguagem C padrão faz muitas suposições que não se aplicam aos sistemas embarcados.

2.3. TEORIA DE *HARDWARE*

2.3.1. MICROCONTROLADOR

Um microcontrolador é um computador, em um chip otimizado para controlar dispositivos eletrônicos. É uma espécie de microprocessador, com memória e interfaces de E/S(I/O) integrados, enfatizando a auto-suficiência, em contraste com um microprocessador de propósito geral, o mesmo tipo usado nos computadores, que requer chips adicionais para prover as funções necessárias [1].

A partir do surgimento dos circuitos integrados TTL, é possível delinear três gerações na implementação dos controladores.

Na primeira geração, as maiorias dos projetos envolviam os CI's TTL, com alto consumo de energia, grande quantidade de chips e dificuldade em se fazer à reengenharia.

A segunda geração ocorre com o advento dos microprocessadores, onde funções até então implementadas por *hardware*, foram substituídas por *software*.

A terceira geração veio para integrar em um único chip grande parte da estrutura. Microcontroladores integram as funções de um microprocessador, memória de dados e de instruções, e alguns mais complexos integram portas seriais, portas paralelas, conversores A/D, timers, *watchdog* e outros.

Os microcontroladores são componentes utilizados em muitos tipos de equipamentos eletrônicos, sendo a grande maioria entre os chips vendidos. Cerca de 50% são controladores "simples", outros 20% são processadores de sinais digitais (DSPs) mais especializados. Os microcontroladores podem ser encontrados em máquinas de lavar, forno de microondas, telefones, etc [1].

2.3.1.1 Microcontrolador Rabbit

O microprocessador Rabbit é produzido pela Z-World que é a pioneira na fabricação de sistemas industriais embarcados.

O microprocessador RCM 3000 agrega processador de alto desempenho, conector Ethernet, memórias RAM e flash, cristal e disponibiliza pinos de entrada e saída.

Características

- 3.3 V para operação;
- Microprocessador Powerful Rabbit 3000;
- Baixo-EMI (típico <10 dB uV/m @ 3 m);
- Ponto Ethernet
- 512 Flash/512 SRAM;
- 52 I/O digitais;
- 6 portas seriais;
- Modos de baixa energia.

Abaixo nas figuras 1 e 2, são apresentados o Microcontrolador *Rabbit* RCM 3000 e o kit de desenvolvimento *Rabbit*.



Figura 1 - Microcontrolador Rabbit RCM3000

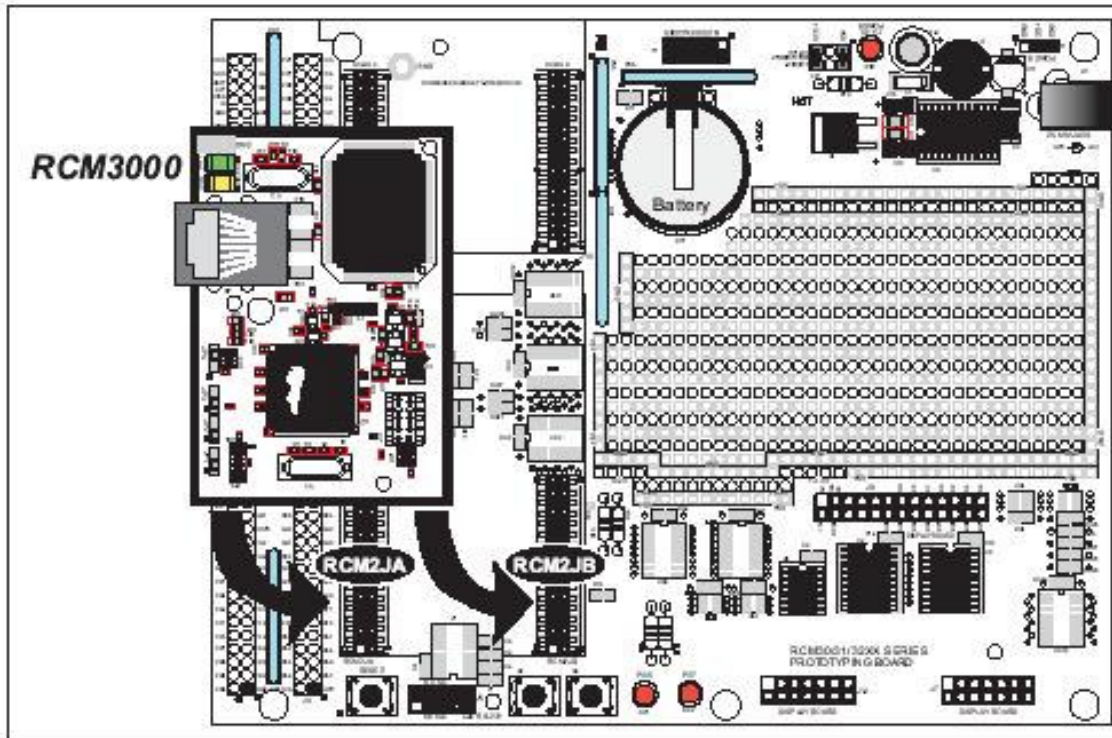


Figura 2 – Kit de desenvolvimento Rabbit

2.3.2. Leitor de Código de Barras

Existem basicamente 4 tipos de leitores de códigos de barras são eles os leitores esfereográficos, leitores a laser, leitores CCD e os leitores com câmeras.

Os leitores estereográficos consistem em um fotodiodo e uma fonte de luz, O fotodiodo mede a intensidade da luz refletida a partir da fonte e gera uma onda para medir o tamanho das barras e os espaços no código.

Os leitores a laser funcionam semelhantemente aos leitores esfereográficos, porém utilizam raio laser ao invés de fonte de luz e também utilizam espelhos para dirigir o raio a todo o código.

Os leitores CCD utilizam uma matriz com centenas de sensores de luz alinhados na ponta do leitor, sendo que cada sensor mede a intensidade de luz recebida.

Os leitores com câmeras utilizam uma pequena câmera que captura imagem do código de barras. Utiliza-se de técnicas de processamento de imagens para decodificar os códigos de barras.

No projeto será utilizado o leitor de código de barras esfereográficos.

2.3.3. Display LCD

Os módulos LCD são interfaces de saída muito útil em sistemas microprocessados. Estes módulos podem ser gráficos e a caracter. Os módulos LCD gráficos são encontrados com resoluções de 122x32, 128x64, 240x64 e 240x128 dots pixel, e geralmente estão disponíveis com 20 pinos para conexão. Os LCD comuns (tipo caracter) são especificados em número de linhas por colunas [2].

Na tabela 1 podem-se ver os modelos de displays existente no mercado e suas relações entre linhas, colunas e pinos.

TABELA 1 - LCD linhas x Colunas x Pinos

Números de Linhas	Números de Colunas	Quantidade de Pinos
8	2	14
12	2	14/15
16	1	14/16
16	2	14/16
16	4	14/16
20	1	14/16
20	2	14/16
20	4	14/16
24	2	14/16
24	4	14/16
40	2	16
40	4	16

Os módulos podem ser encontrados com *LED backlight* (com uma iluminação de fundo) para facilitar as leituras durante a noite.

A corrente de alimentação deste *led* varia de 100 a 200 mA, dependendo do modelo.

Estes módulos utilizam um controlador próprio, permitindo sua interligação com outras placas através de seus pinos, onde deve ser alimentado o módulo e interligado o barramento de dados e controle do módulo com a placa do usuário [2].

Além de alimentar e conectar os pinos do módulo com a placa do usuário deverá haver um protocolo de comunicação entre as partes, que envolve o envio de bytes de instruções e bytes de dados pelo sistema do usuário.

Neste projeto, a princípio será utilizado um módulo LCD comum de 2 linhas e 16 colunas para mostrar a saída de dados.

A tabela 2 abaixo descreve cada pino do módulo ou do display para conexão deste a outras placas:

TABELA 2 – Descrições e funções dos pinos

Pino	Função	Descrição
1	A	Iluminação +
2	K	Iluminação -
3	VSS	Terra
4	VDD	Alimentação
5	VEE	Ajuste Contraste
6	RS	1 – Dado, 0 Instrução
7	R/W	1 -Escrita/0 - Leitura
8	E	1- Habilitado/0- Desabilitado
9	DB0	Linha de Dados 0
10	DB1	Linha de Dados 1
11	BD2	Linha de Dados 2

12	DB3	Linha de Dados 3
13	DB4	Linha de Dados 4
14	DB5	Linha de Dados 5
15	DB6	Linha de Dados 6
16	DB7	Linha de Dados 7

Na figura 3 segue o display a ser utilizado nesse projeto de 2 linhas por 16 colunas.

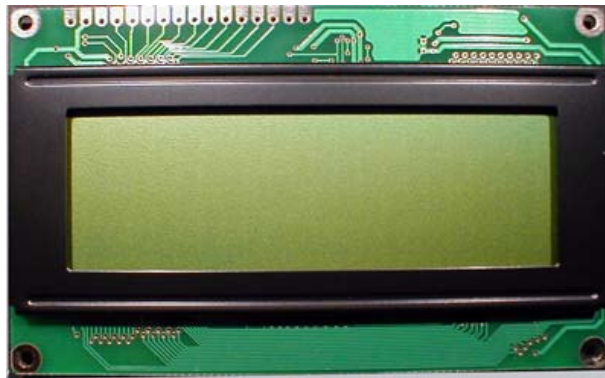


Figura 3 – Display LCD 16 x 2

2.3.4. Ethernet

A Ethernet é uma interconexão de redes locais baseado no envio de pacotes. Com relação aos aspectos de redes locais define tanto a parte lógica (protocolos, pacotes, etc.) quanto na parte física (sinais elétricos, cabeamento, etc.)[1].

Com relação a redes locais é o padrão mais implementado no mundo.

Abaixo na figura 4 pode ser visualizado um exemplo de uma rede ethernet.

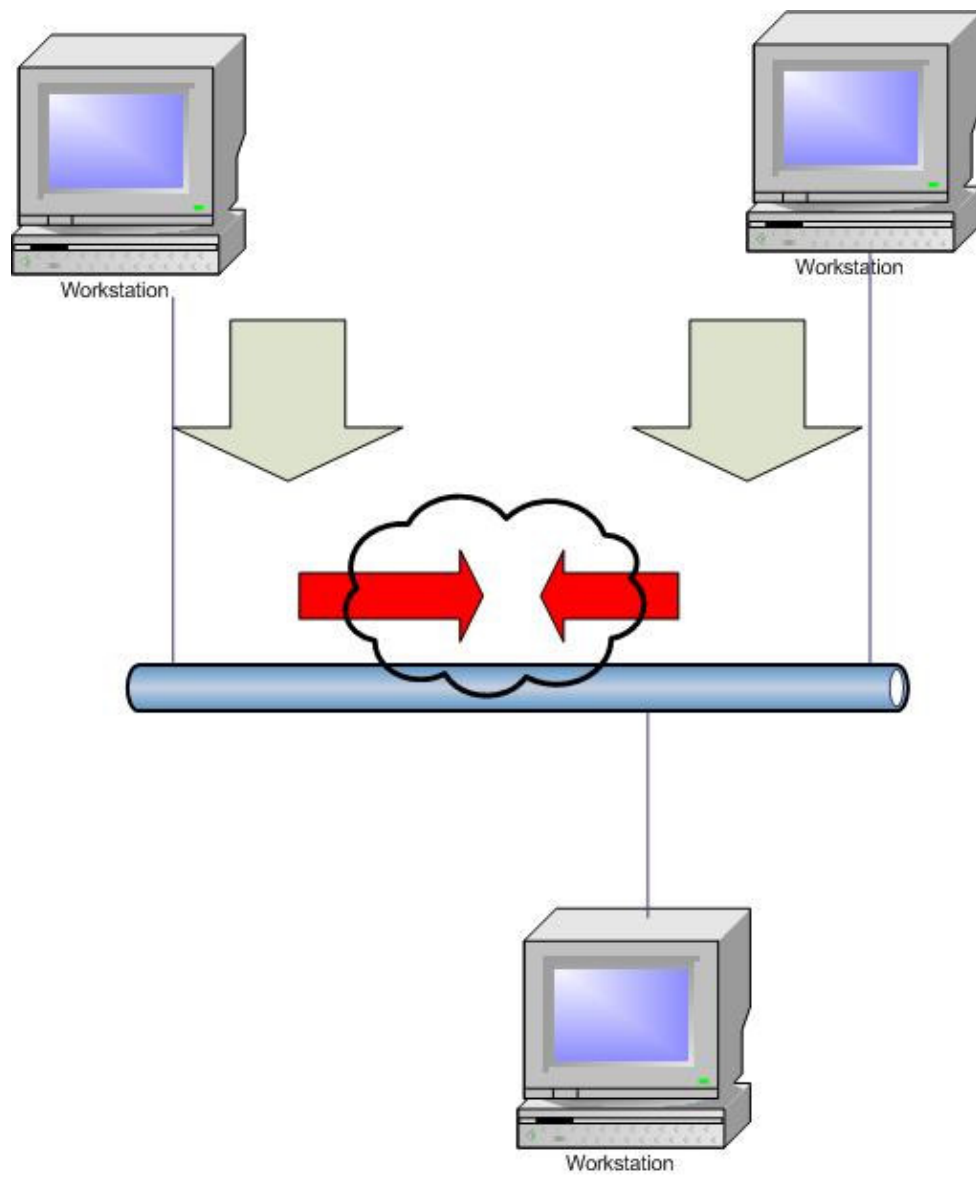


Figura 4 – Rede Ethernet

3. ESPECIFICAÇÃO TÉCNICA

Neste capítulo será apresentada uma visão geral sobre o sistema desenvolvido. Está dividido entre hardware, software, aplicabilidades e funcionalidades de hardware e software.

3.1. ESPECIFICAÇÃO DO HARDWARE

O sistema a ser desenvolvido é mostrado abaixo no diagrama de blocos do hardware na figura 5, é composto por um microcontrolador, memórias, um display LCD, botões e leitor de código de barras.

A geladeira inteligente tem por função controlar a entrada e saída dos seus alimentos para os mais variáveis fins e também disponibilizar através de uma página de Internet uma lista dos alimentos contidos na geladeira para serem disponíveis aos seus usuários de qualquer lugar.

O *hardware* consiste basicamente na aquisição de dados, processamento dos dados e comunicação com o pc através da Ethernet e saída dos dados.

Para a aquisição de dados será utilizado um leitor de código de barras que é acoplado ao microprocessador, para poder ler os códigos de barras dos produtos e botões/teclados para configurações.

Nesse sistema será utilizado o microprocessador Rabbit 3000, com seu kit de desenvolvimento, que possui algumas funções já mapeadas que facilitando o desenvolvimento. O Rabbit 3000 é programado em linguagem C através do programa Dynamic C, que suporta versões anteriores dos processadores e também possui facilidades como o debug em tempo real, que facilita para o programador.

Para armazenamento de dados será utilizado as memórias flash que fazem parte do microprocessador Rabbit 3000, nela serão armazenados os produtos lidos e também os produtos como num mini banco de dados.

A comunicação com o PC será feito através da conexão Ethernet, cuja é umas das funcionalidades inclusas no Kit do Rabbit 3000.

Para a saída dos dados, será utilizado um display LCD para que sejam mostrado o nome do produto na hora em que é lido.

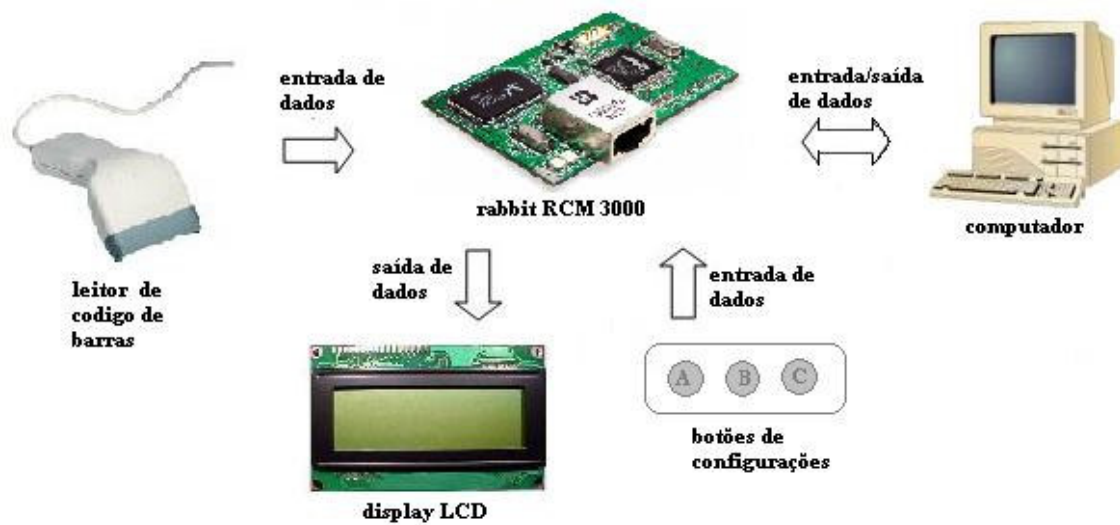


FIGURA 5 – diagrama de blocos do hardware

3.2. ESPECIFICAÇÃO DO SOFTWARE

Nesse sistema a linguagem de programação escolhida é a linguagem C utilizando o software Dynamic C, devido a sua grande eficiência, praticidade, e funcionalidades.

Também foi utilizado as linguagens HTML e CGI, a primeira foi usado devido ao fato da utilização da página da *web* e o CGI foi necessário uma vez que a página *web* deveria ser dinâmica, pois deveria ter atualização quase que em tempo real. Assim sendo, como a página está alocada no microcontrolador Rabbit utiliza-se o cgi no programa principal, de forma que possa transformar as variáveis de produtos e de quantidades, da linguagem C para o formato CGI, dessa forma já na forma CGI é possível utilizá-los na página HTML, assim foi possível ter atualizados a listas dos produtos contidos na geladeira.

Abaixo na figura 6 é mostrado aonde o software será visualizado, através do protocolo *tcp/ip* a página *web* é visualizado por um computador que esteja conectado pelo cabo *ethernet*.



FIGURA 6 – Conexão TCP/IP com o microcomputador

3.3. APLICABILIDADE / FUNCIONALIDADE DO HARDWARE

O processamento dos dados lidos pelo leitor de código de barras será feito da seguinte forma:

- O programa fica num laço a espera da leitura de um código de um produto através do leitor de código de barras.
- Em seguida verifica-se numa estrutura de dados, que é onde estão alocados os dados dos produtos cadastrados, se estiver cadastrado e a quantidade estiver zero, o produto é automaticamente adicionado à lista de produtos contidos na geladeira.
- Se o produto lido estiver cadastrado e sua quantidade estiver diferente de zero, no display aparecerão as opções de inserir e remover, sendo feitos através dos botões.
- Se o produto lido não estiver cadastrado, aparecerá no display a opção de cadastrá-lo, e para se fazer o cadastro é necessário o uso dos botões sendo os dois primeiros, um para incrementar a letra e o outro para decrementar a letra, começando a partir da letra 'A', o terceiro para avançar para a próxima coluna, o quarto botão para limpar e o quinto botão é o *enter*, que serve para confirmar o cadastro. Ao fim do cadastro é mostrada no display a confirmação se o cadastro foi efetuado com sucesso.
- Para se remover, o programa fica num laço a espera da leitura de um código de barras e ao invés de escolher a opção de inserir, escolhe-se para remover, essa opção só estará disponível desde que a quantidade seja maior que zero.

Isso pode ser verificado através de uma página WEB dinâmica acessando o endereço IP definido no programa para o microcontrolador Rabbit.

A funcionalidade pode ser verificada no fluxograma abaixo na figura 7.

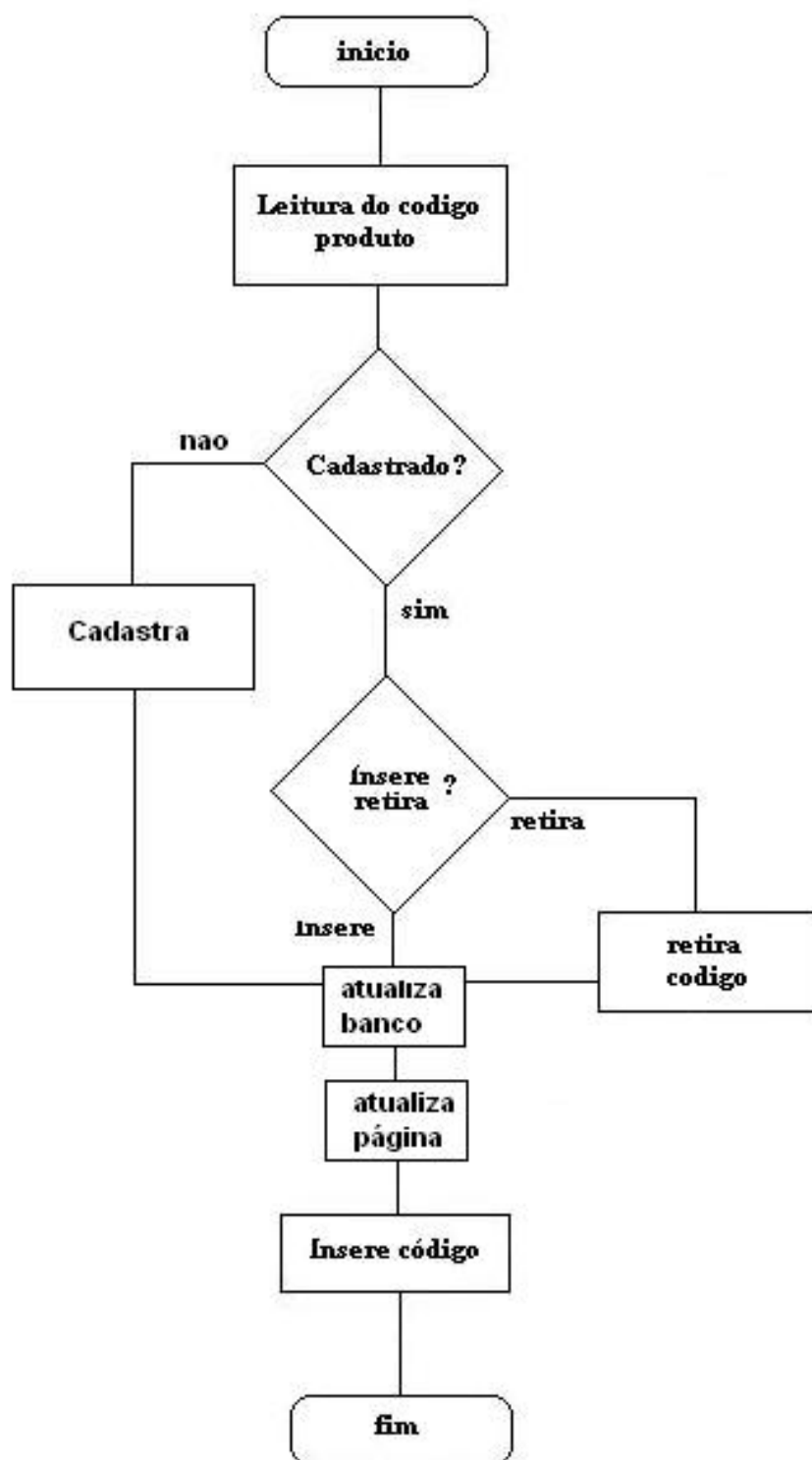


FIGURA 7 – Funcionalidade do hardware do sistema

3.4. APLICABILIDADE / FUNCIONALIDADE DO SOFTWARE

O software é desenvolvido em HTML e funciona rodando dentro do microprocessador *Rabbit*. Basicamente tem como função disponibilizar para os usuários a lista de produtos contidos na geladeira.

Para que os dados tenham atualização quase real, é necessário utilizar o CGI, uma vez que o HTML é estático, dessa forma o CGI transforma as variáveis de produto e quantidade da linguagem C para CGI, através de uma função e de algumas configurações no software.

Os produtos e seus respectivos códigos serão cadastrados por um usuário, mas poderia ser conectado, por exemplo, a algum fornecedor ou supermercado para que se obtivesse os produtos e códigos.

4. PROJETO

Neste capítulo é apresentado de forma mais detalhado o projeto de software e de hardware.

4.1. PROJETO DE HARDWARE

O projeto de Hardware consiste no microcontrolador Rabbit ligado ao display LCD de 16x2 e também a um leitor de código de barras, ao ler o código de barras é adicionado ou retirado o produto da memória e mostrado no display LCD.

4.1.1. Display LCD

Na tabela 3 abaixo é mostrada a descrição dos sinais do display LCD e na figura a seguir é mostrado as conexões entre o LCD e a PORT A do Kit do microprocessador Rabbit e na figura 8 pode-se ver as ligações entre as portas 'A' do *Rabbit* e o display LCD.

Tabela 3 - descrição dos Sinais do display LCD

Pino	Sinal	Descrição	Tipo	Conexao
1	A	Iluminação +	Entrada	Não Conectado
2	K	Iluminação -	Entrada	Não Conectado
3	VSS	Terra	Entrada	Conectada a 0V
4	VDD	Alimentação	Entrada	Conectada a 5V
5	VEE	Ajuste Contraste	Entrada	Conectada a 0V
6	RS	Registrador	Entrada	Conectado ao PA6
7	R/W	Escrita/Leitura	Entrada	Conectada a 0V
8	E	Habilitação	Entrada	Conectado ao PA7
9	DB0	Linha de Dados 0	Entrada	Não Conectado
10	DB1	Linha de Dados 1	Entrada	Não Conectado
11	DB2	Linha de Dados 2	Entrada	Não Conectado
12	DB3	Linha de Dados 3	Entrada	Não Conectado
13	DB4	Linha de Dados 4	Entrada	Conectado ao PA0
14	DB5	Linha de Dados 5	Entrada	Conectado ao PA1
15	DB6	Linha de Dados 6	Entrada	Conectado ao PA2
16	DB7	Linha de Dados 7	Entrada	Conectado ao PA3

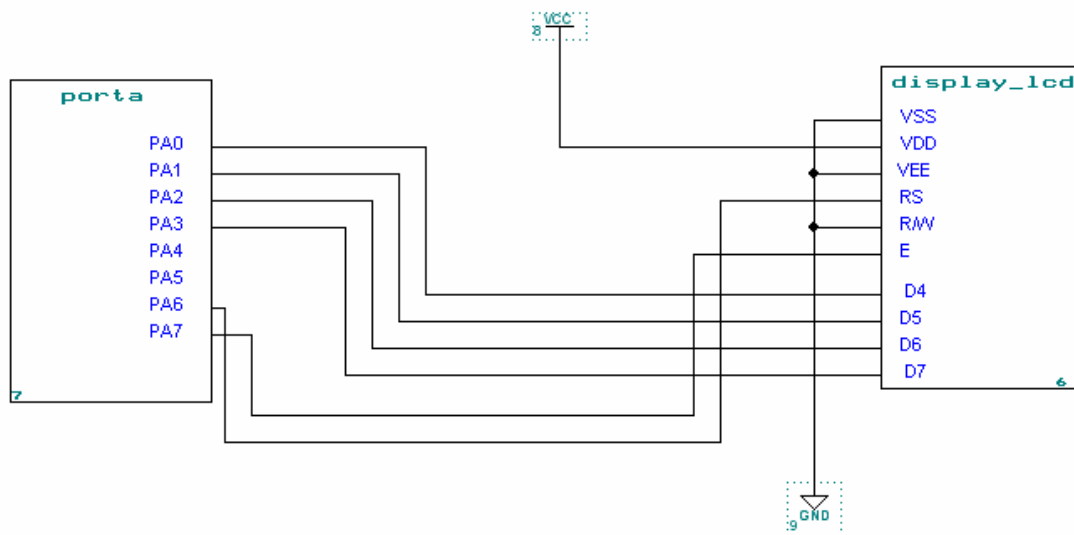


FIGURA 8 - Esquemático de interligação entre Rabbit e display LCD.

4.1.2. Leitor de Código de Barras

O leitor de códigos de barras é conectado ao microprocessador Rabbit através do PORT C, onde o RX e o TX são conectados respectivamente nos TBX e RBX (PC4 e PC5).

Na tabela 4 pode ser visualizadas os sinais do leitor de código de barras e suas conexões e na figura 10 pode ser visualizada a conexão entre o microcontrolador *Rabbit* e o leitor de código de barras.

Tabela 4 - Descrição dos sinais do leitor de código de barras

Pino	Sinal	Descrição	Tipo	Conexão
2	TX	Transmissão	Entrada	Conectado ao PC4 (RBX)
3	RX	Recepção	Saída	Conectado ao PC5 (TBX)
5	GND	Terra	Entrada	Conectado a 0V



FIGURA 09 – Esquemático de interligação entre Rabbit e o leitor de código de barras.

4.1.3. Protocolo de Comunicação

O protocolo de comunicação utilizado foi o TCP/IP, que são base de aplicações disponibilizadas e necessárias à Internet.

O protocolo IP, camada de rede (3), envia dados para rede sem a preocupação de verificar a chegada dos respectivos datagramas. Os protocolos da camada acima, *host-host* ou transporte (4), especificamente TCP, definem a maneira para tratar datagramas perdidos ou corromptos. Além disto, TCP é responsável pela segurança na transmissão/chegada dos dados ao destino e também define todo o processo de início de conexão e multiplexação de múltiplos protocolos da camada de aplicação (7) em uma única conexão, otimizando assim a conexão múltipla de aplicações com o mesmo destino.

O protocolo TCP é orientado a conexão sendo isto claramente observado no processo de inicialização da conexão. O TCP aplica o algoritmo *three-way handshake* ou *three-fold* nesta inicialização.

4.1.4. Kit do Microprocessador Rabbit

Como o kit do microprocessador rabbit não será necessário montar, já é fornecido pronto para usar, na figura 11 é mostrado o esquemático do Rabbit , onde pode ser visto os componentes utilizados pelo kit.

Os conectores J1 e J2 serão utilizados, como podem ser vistos na figura 10, pois é onde se encontram as saídas dos PORT's utilizados tanto para o leitor de código de barras quanto para o display LCD.

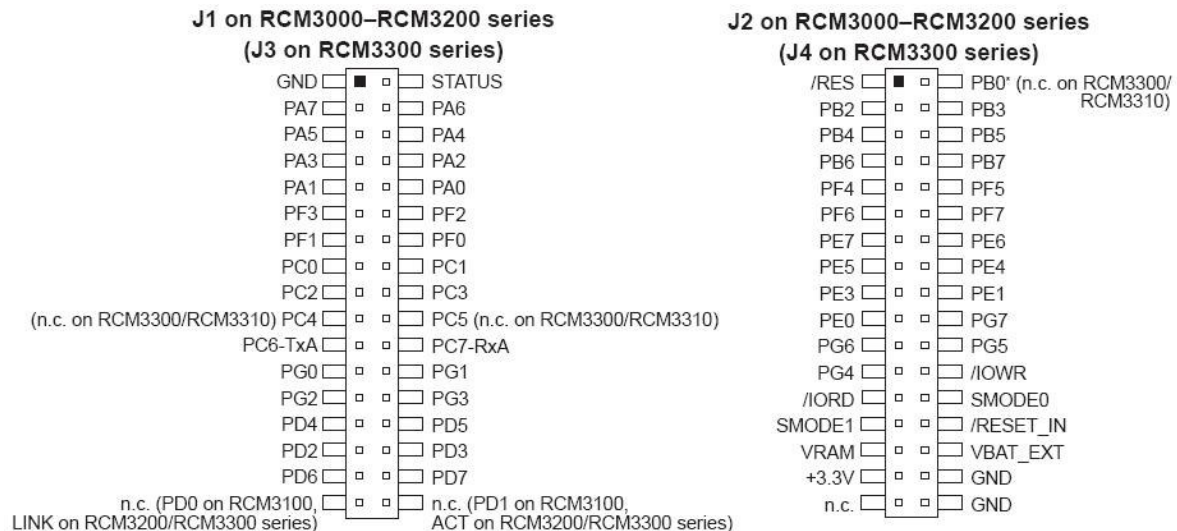


FIGURA 10 – Pinagens do Rabbit (conectores J1 e J2)

4.2. PROJETO DE SOFTWARE

O software tem por objetivo prover acesso ao conteúdo dos dados contidos na geladeira através da página HTML com CGI que é rodado de dentro do microcontrolador *Rabbit*, ou seja, provém acesso aos dados dos produtos para os usuários que além de acrescentar dados na tabela, pode remover os dados. Na figura 11 é mostrado o diagrama de casos de uso e em seguida tem a descrição dos casos de uso.

4.2.1. Diagrama de Caso de uso

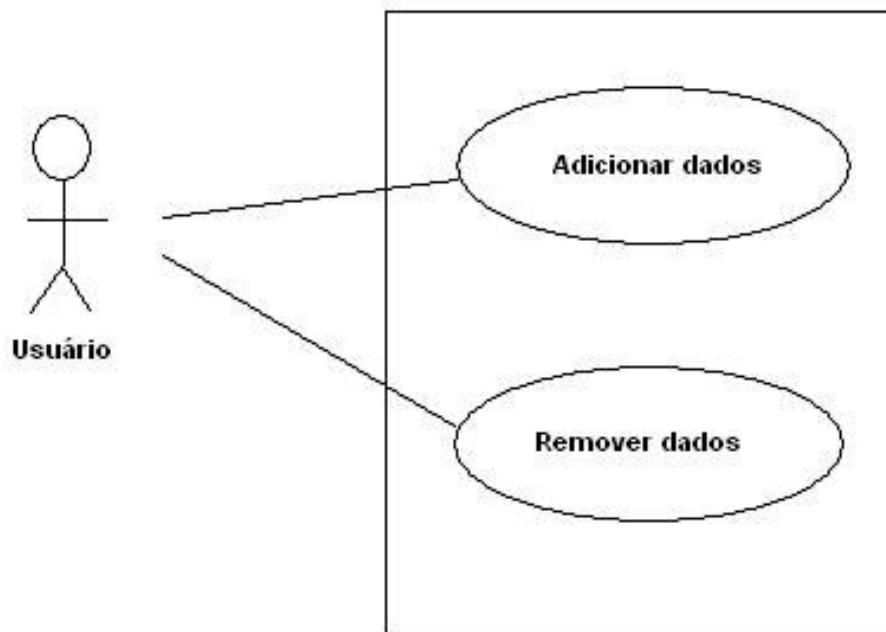


FIGURA 11 - Diagrama de caso de uso

4.2.2.Descrição de casos de uso

Segue a descrição dos casos de usos.

Casos de uso: visualizar dados.

Ator: Usuário.

Descrição: Ator visualiza os dados contidos na geladeira através do site.

Tipo: secundário

Propósito: visualizar os dados contidos.

Casos de uso: adicionar dados.

Ator: Usuário.

Descrição: Usuário adiciona os produtos à lista de dados.

Tipo: primário

Propósito: inserir novos dados.

Casos de uso: remover dados.

Ator: Usuário.

Descrição: Usuário remove os produtos à lista de dados.

Tipo: primário

Propósito: remover dados.

4.2.3. Diagrama de seqüência

Na figura 12 é mostrado o diagrama de seqüência de visualizar dados.

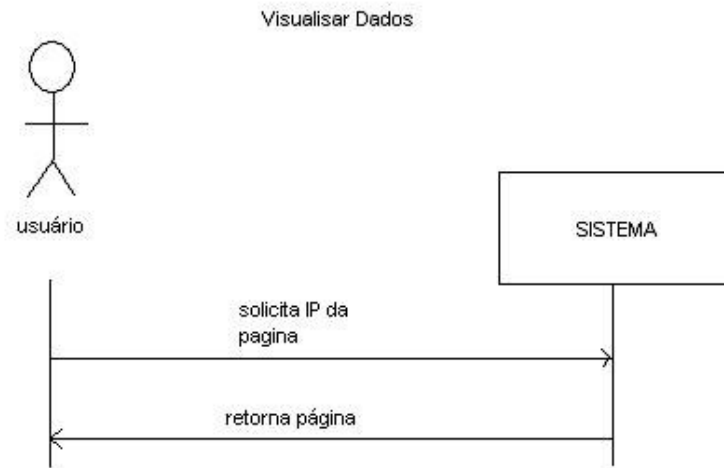


FIGURA 12– Diagrama de seqüência visualizar dados

Na figura 13 é mostrado o diagrama de seqüência de Adicionar dados, o usuário adiciona o produto e recebe uma confirmação de que a operação foi executada com sucesso ou não.

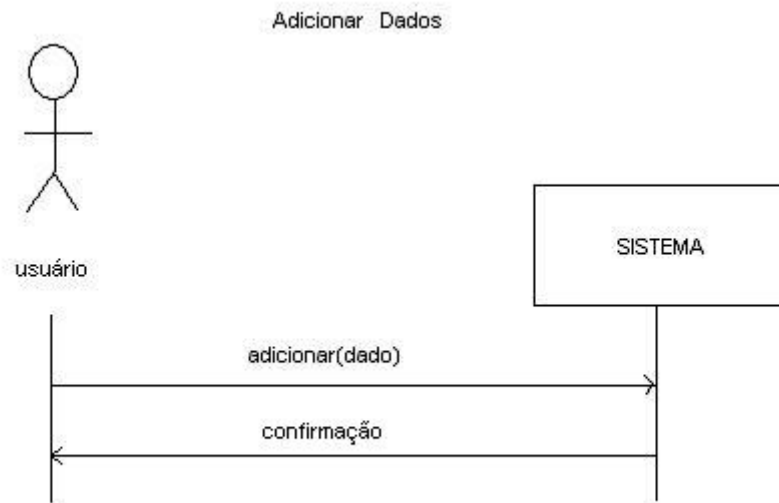


Figura 13– Diagrama de seqüência adicionar dados

Na figura 14 é mostrado o diagrama de seqüência de Remover dados, o usuário remove o produto e recebe uma confirmação de que a operação foi executada com sucesso ou não.

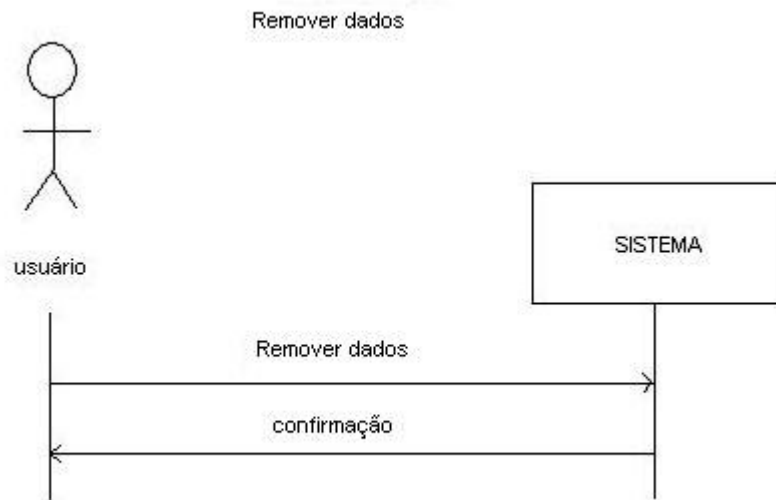


Figura 14– Diagrama de seqüência remover dados

4.2.4. Estrutura para armazenamento de dados

Na tabela 5 é mostrada a estrutura de tabela dos dados, que são os como os dados estão dispostos no programa. Os códigos que são os códigos de barras e o produto, que é a descrição dos dados, ficam armazenados em uma string, já a quantidade de cada produto fica armazenada num inteiro.

Tabela 5 - Estrutura de dados

Código	String
Produto	String
Quantidade	Int

5. CRONOGRAMA DO PROJETO

Tabela 6 – Cronograma do projeto

Tarefas	Março	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro	Novembro
1 Elaboração da Revisão Bibliográfica									
2 Elaboração da Especificação									
3 Testes com o microprocessador									
4 Elaboração do programa do microprocessador									
5 Testes dos módulos com o microprocessador									
6 Elaboração dos Software									
7 Integração Software/Hardware									
8 Revisão da Documentação									
9 Conclusão do projeto									

6. VALIDAÇÃO

Para a validação do projeto foram realizados testes sobre os módulos em separados para verificar o funcionamento de cada um deles.

A avaliação geral do projeto foi feita com todos os módulos integrados, lendo produtos já cadastrados no leitor de códigos de barras e também produtos não cadastrados, sendo assim foi feito o cadastro e verificado se estava funcionando através da página de *HTML* que rodava de dentro do *rabbit*, podendo ser verificado através do endereço de IP, uma vez que o *rabbit* possuía o endereço de IP 10.10.6.102 e o Computador possuía o endereço de IP 10.10.6.101, então num Internet Explorer foi digitado o endereço do IP do *rabbit* e verificado as alterações em tempo real.

7. ESTUDO DE VIABILIDADE TÉCNICO-ECONÔMICA

Abaixo será apresentado na tabela 7 de estimativa de custo do projeto, estão sendo descritos todos os recursos necessários, inclusive os softwares utilizados na faculdade e seus respectivos preços em reais.

Tabela 7 – Estimativa de custo do Projeto

Recurso/Material	Quantidade	Custo Unitário (R\$)	Custo (R\$)
Microprocessador Rabbit RCM 3000	1	210,00	210,00
Kit de desenvolvimento	1	1000,00	1000,00
Leitor de código de barras	1	190,00	190,00
Microsoft Windows 2000	1	467,00	467,00
Microsoft Office XP	1	406,00	406,00
Display LCD	1	38,00	38,00
Componentes diversos	1	150,00	150,00
Horas de desenvolvimento	660	10,00	6600,00
		Total	9061,00

8. RESULTADOS E DISCUSSÃO

Os resultados obtidos foram satisfatórios, pois foi possível visualizar o projeto funcionando conforme proposto na definição do projeto, os módulos apesar das dificuldades encontradas no início do desenvolvimento, funcionaram o esperado.

Os Produtos eram lidos corretamente pelo leitor códigos de barras e processados corretamente pelo microprocessador *Rabbit* e visualizado numa página de web, através do endereço 10.10.6.102 que foi definido no programa.

Através da página de html, era possível ver em tempo quase real o que estava sendo adicionado ou retirado da geladeira.

Abaixo é mostrada na figura 15 a foto do projeto montado.

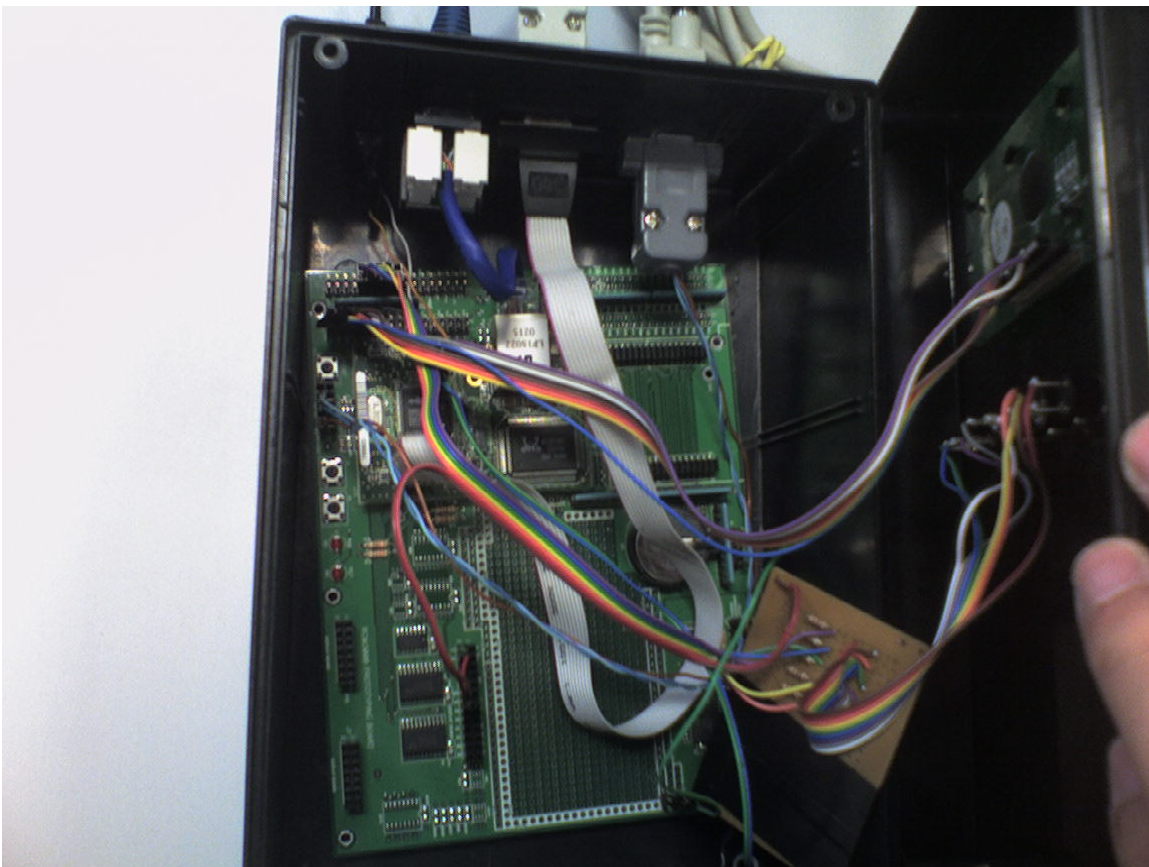


FIGURA 15 - Projeto Mostrado internamente montado com o kit microcontrolador rabbit

Na figura 16 são mostradas as conexões dos periféricos com o microcontrolador *rabbit*, o primeiro é ligado o leitor de código de barras, segundo é cabo de programação do *rabbit*, terceiro a rede ethernet e o quarto de alimentação.

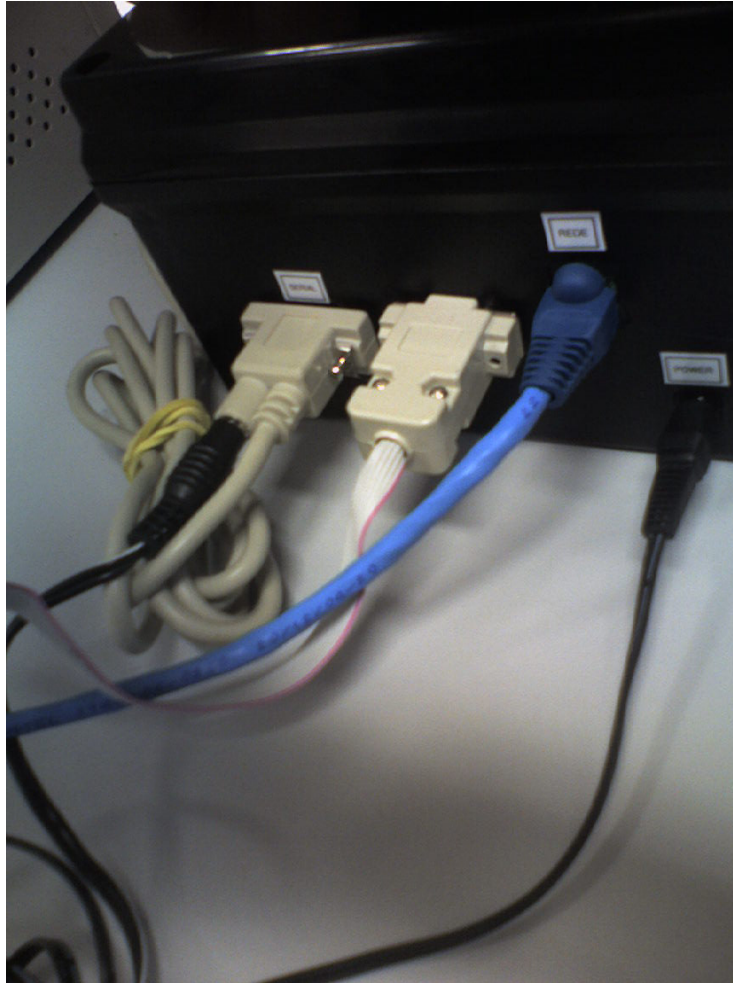


FIGURA 16 – As conexões dos periféricos ao projeto

Já na figura 17 é mostrada a foto vista de frente do projeto, podendo ser visto o display LCD e também os botões que auxiliam no funcionamento do projeto, uma vez que servem para definir alguns parâmetros, como por exemplos se o objeto está entrando ou saindo, ou mesmo cadastrando um produto que não esteja cadastrado.



FIGURA 17 – Parte do Projeto montado com display, botões.

Na figura 18 é apresentada a página que roda dentro do microprocessador *Rabbit* através do endereço 10.10.6.102. A cada dois segundos a página é atualizada, uma vez que é necessário que haja quase que em tempo a atualização dos dados que entraram e saíram da geladeira.

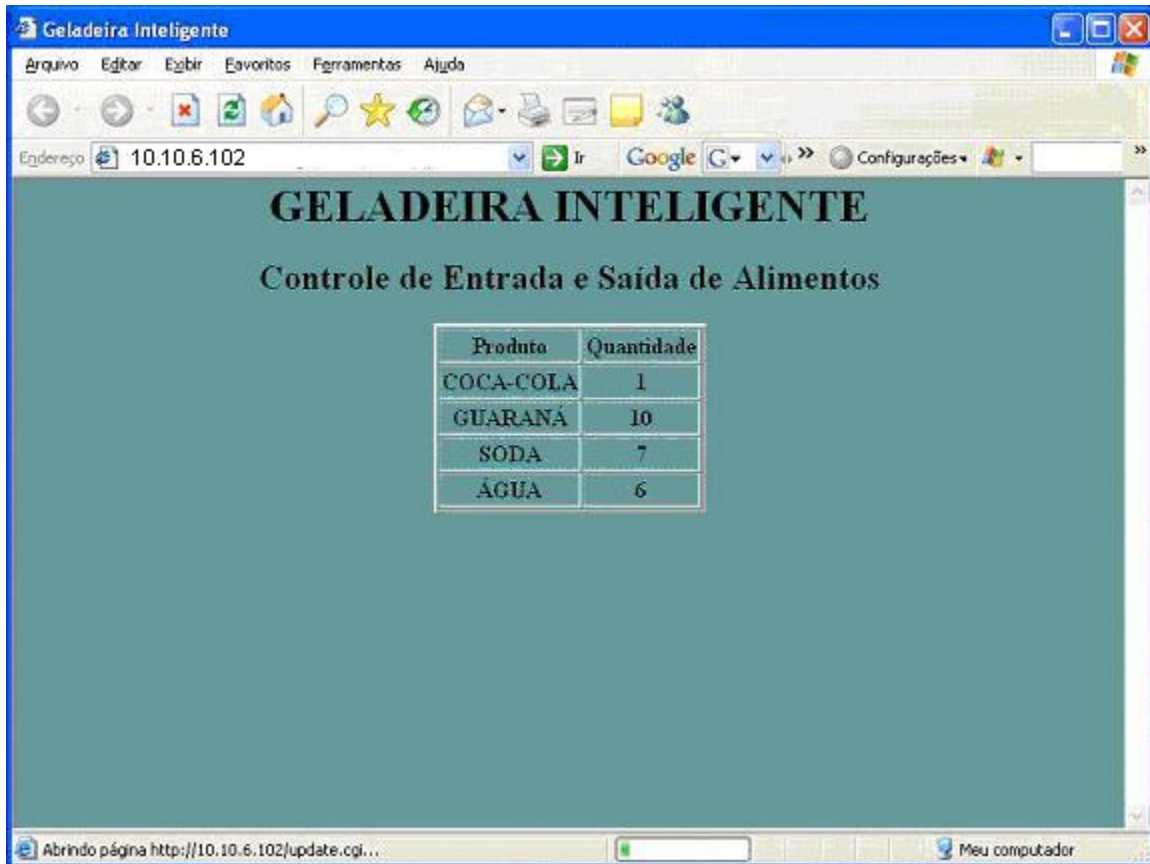


FIGURA 18 – Página da Web do projeto

9. CONCLUSÃO

Os resultados obtidos foram satisfatórios, contudo não são suficientes para torná-lo um produto comercial, isto se deve ao fato dos módulos serem grandes, como pode ter visto anteriormente. Porém a idéia do projeto em geral, com intensos testes, utilizando uma tecnologia mais otimizada para que se possa integrar a uma geladeira, um banco de dados externo e também se oferecendo mais opções no site, por exemplo, como vendas, busca de produtos mais baratos e etc.

Dessa forma o projeto foi realizado cumprido com o que era previsto, apesar das dificuldades encontradas no seu desenvolvimento, principalmente pelo fato de estar a um ano sem estudar, uma vez que não estudei no ano de 2005 por motivos particulares.

Outro problema encontrado foi no recebimento dos dados lidos pelo leitor de código de barras e como tratá-los. Também foi necessário avaliar como fazer uma página dinâmica dentro do microprocessador *Rabbit*.

Além disso, foi encontrado um outro grande problema no desenvolvimento da página HTML, uma vez que a página tinha que ser dinâmica para que fosse atualizados os produtos e quantidades da lista contidas na geladeira, toda a vez que o programa ficava num laço esperando alguma ação, a página travava quando acessada no *browser* de Internet, pois a página tentava atualizar o cgi e não conseguia porque o programa não estava pronto pra responder a solicitação, então foi resolvido ao adicionar a função `http_handler()` em cada laço dentro do programa e o problema foi resolvido. Sendo assim, toda vez que era acessada, essa função respondia a solicitação não deixando travar.

Contudo, o projeto foi de grande ajuda para o meu crescimento profissional, uma vez que foram estipulados prazos, metas, desafios e os problemas iam surgindo e sempre eram necessários decisões rápidas para solucioná-los como no ambiente de trabalho.

10. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] COMER, Douglas E., Redes de Computadores e Internet. São Paulo Prentice-Hall,1999.
- [2] Manual do display LCD
<http://planeta.terra.com.br/educacao/mdmau/LCD.PDF>
- [3] Código de barras sem mistérios por Luiz Cláudio C. V. da Rocha
www.microsoft.com.br/brasil/msdn/
- [4] American National Standards Institute, *American National Standard for Information Systems­Programming Language C*, X3.159-1989.
- [5] www.eanbrasil.com.br
- [6] Z-World, RabbitCore RCM3000 Getting Started Manual. Califórnia,2005
- [7] www.rabbitsemicondutor.com